

[illegible]

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

```
1 0001 0 MODULE SEPARATE ( %TITLE 'Print Symbiont -- separation routines'
2 0002 0 IDENT = 'V04-001',
3 0003 0 ADDRESSING_MODE (EXTERNAL = GENERAL)
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 ++
33 0033 1 FACILITY:
34 0034 1 Print Symbiont.
35 0035 1
36 0036 1 ABSTRACT:
37 0037 1 File and module input routines.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1 VAX/VMS user mode.
41 0041 1 --
42 0042 1
43 0043 1 AUTHOR: Rowland R. Bradley
44 0044 1
45 0045 1 CREATION DATE: April 1, 1984
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1 40-001 RRB0006 Rowland R. Bradley 14-Aug-1984
50 0050 1 Add a <CR> to the page header. This fixes QAR 0682
51 0051 1 QAR 1737, loss of first line of data of every page.
52 0052 1
53 0053 1 3B-005 RRB0005 Rowland R. Bradley 02-Aug-1984
54 0054 1 Remove the phrase "in this file" in (file) record description.
55 0055 1 Remove extra spaces in job description after start date
56 0056 1 and after queued to date. Slightly alter filename algorithm to
57 0057 1 avoid placing long filename on a single line and NOT displaying
```



```
58 0058 1 | the type and version (or vice versa). Add a call to
59 0059 1 | PSMSREAD_ITEM_DX in GET_QUALIFIERS to test correctness(this
60 0060 1 | is my call and will remain as a test). Modify the file trailer
61 0061 1 | page to avoid an ugly truncation of the input file. Add the
62 0062 1 | acronym "UIC" to the job description sentence. Add /FEED and
63 0063 1 | /NOFEED to the list of qualifiers in GET_QUALIFIERS.
64 0064 1 |
65 0065 1 |
66 0066 1 | 3B-004 RRB0004 Rowland R. Bradley 04-May-1984
67 0067 1 | Avoid truncation of the Job Flag Page when burst pages
68 0068 1 | are not specified.
69 0069 1 |
70 0070 1 | 3B-003 GRR0003 Gregory R. Robert 29-Apr-1984
71 0071 1 | Removed reference to JBC$_JOBREQUEUE until job controller
72 0072 1 | message definitions appear in LIB.L32. Changed ABORTED
73 0073 1 | and REQUEUED words in job sentence to uppercase for emphasis.
74 0074 1 | Considered left and right margins in computing page header
75 0075 1 | size. Create page header only once per task. Fix page header
76 0076 1 | to have file revision date instead of revision number.
77 0077 1 |
78 0078 1 | 3B-002 RRB0002 Rowland R. Bradley 27-Apr-1984
79 0079 1 | Add dynamic Page Header routine, display all queue qualifiers
80 0080 1 | in a single phrase, fix form feed and line one problems, print
81 0081 1 | /setup_file & /setup_form, insert job number in burst chars,
82 0082 1 | FIX: footer bar, devicename, receipt box, file desc sentence,
83 0083 1 | trailer header bar, digital logo dynamics, /width,
84 0084 1 | psm$announce display size, fab valid bit, page setup quals.
85 0085 1 |
86 0086 1 | 3B-001 RRB0001 Rowland R. Bradley 01-Apr-1984
87 0087 1 | Original version
88 0088 1 |
89 0089 1 | **
```


SEPARATE
V04-001

Print Symbiont -- separation routines

G 13
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 3
(2)

```
: 91      0090 1 LIBRARY 'SYSS$LIBRARY:LIB';
: 92      0091 1 REQUIRE 'LIB$:SMBDEF';
: 93      0583 1 REQUIRE 'SRC$:SMBREQ';
: 94      1040 1
: 95      1041 1 EXTERNAL ROUTINE
: 96      1042 1     PSMSBANNER,
: 97      1043 1     PSMSREAD_ITEM_DX      ! strictly to test behavior
: 98      1044 1
: 99      1045 1
: 100     1046 1 LITERAL
: 101     1047 1     UPCASE_MASK= %B '0100000',      ! lowercase to uppercase
: 102     1048 1     LEAD_MASK  = %B '001000',      ! delete leading blanks
: 103     1049 1     SIZE      = 0,
: 104     1050 1     ADDR      = 1,
: 105     1051 1
: 106     1052 1
: 107     1053 1 OWN
: 108     1054 1     BURST
: 109     1055 1
```

111	1056	1	FORWARD ROUTINE		
112	1057	1	PSMSFILE-BURST	:	
113	1058	1	PSMSFILE-FLAG	:	
114	1059	1	PSMSFILE-TRAILER	:	
115	1060	1	PSMSJOB-BURST	:	
116	1061	1	PSMSJOB-FLAG	:	
117	1062	1	PSMSJOB-TRAILER	:	
118	1063	1	PSMSPAGE-HEADER	:	
119	1064	1			
120	1065	1	PARSE FILE NAME	:	
121	1066	1	ALLOCATE PAGE	:	
122	1067	1	DEALLOCATE PAGE	:	
123	1068	1	CREATE PAGE HEADER	:	
124	1069	1	RETURN FRAME LENGTH	:	
125	1070	1	RETURN FRAME WIDTH	:	
126	1071	1	FILL JOB FLAG	:	NOVALUE,
127	1072	1	FILL FILE FLAG	:	NOVALUE,
128	1073	1	FILL JOB TRAILER	:	NOVALUE,
129	1074	1	FILL FILE TRAILER	:	NOVALUE,
130	1075	1	GET FORM SIZE	:	NOVALUE,
131	1076	1	GET REVISION DATE	:	NOVALUE,
132	1077	1	GET SYSTEM ANNOUNCEMENT	:	NOVALUE,
133	1078	1	GET JOB DESCRIPTION	:	NOVALUE,
134	1079	1	GET FILE DESCRIPTION	:	NOVALUE,
135	1080	1	GET FILE NAME	:	NOVALUE,
136	1081	1	GET JOB NAME	:	NOVALUE,
137	1082	1	GET EOJ	:	NOVALUE,
138	1083	1	GET EOF	:	NOVALUE,
139	1084	1	GET ACCOUNTING INFO	:	NOVALUE,
140	1085	1	GET QUALIFIERS	:	NOVALUE,
141	1086	1	GET QUEUE QUALIFIERS	:	NOVALUE,
142	1087	1	GET USER NOTE	:	NOVALUE,
143	1088	1	GET RECEIPT BOX	:	NOVALUE,
144	1089	1	GET VMS LOGO	:	NOVALUE,
145	1090	1	GET DIGITAL LOGO	:	NOVALUE,
146	1091	1	GET RULER FINE	:	NOVALUE,
147	1092	1	GET RULER COARSE	:	NOVALUE,
148	1093	1	SCROLL FRAME	:	NOVALUE,
149	1094	1	FILL FRAME	:	NOVALUE,
150	1095	1	MOVE FRAME	:	NOVALUE,
151	1096	1	INSERT FRAME	:	NOVALUE,
152	1097	1	CENTER FRAME	:	NOVALUE,
153	1098	1	MERGE FRAME	:	NOVALUE,
154	1099	1	INSERT NAME BANNER	:	scrolls/fills frame w/ string
155	1100	1	INSERT FILENAME BANNER	:	fills a frame with chars
156	1101	1	INSERT JOBNUMBER BANNER	:	inserts unlimited strings
157	1102	1	DELIMIT STRING	:	inserts delimited strings
158	1103	1	DELIMIT STRING NOT	:	centers unlimited strings
159	1104	1	FIND DEST PTR	:	merges "pure" frames
160	1105	1	FIND SOURCE PTR	:	inserts banners into frames
161	1106	1	DISCARD	:	inserts banners into frames
162	1107	1	FILE_OPEN	:	inserts banners into frames
163	1108	1	:	:	delimits strings
164	1109	1	:	:	delimits strings

```
1110 1 %sbttl 'PSM$FILE_BURST - Print a File Burst Page'
1111 1 Functional Description:
1112 1 This routine controls the creation of the file burst page. The
1113 1 FUNCTION code dictates the action taken in creation.
1114 1 FUNCTION:
1115 1 OPEN - Allocate and create the File Burst Page
1116 1 READ - Return the current line of the File Burst Page
1117 1 CLOSE - Return the buffer allocated on OPEN
1118 1
1119 1 Formal Parameters:
1120 1 SMB_CONTEXT - Pointer to the SMB
1121 1 USER_CONTEXT - User defined pointer (not used here)
1122 1 FUNCTION - OPEN, READ, CLOSE
1123 1 FUNC_DESC - Pointer to functionally dependent descriptor
1124 1 FUNC_ARG - Pointer to functionally dependent argument
1125 1
1126 1 Implicit Inputs:
1127 1 none
1128 1
1129 1 Implicit Outputs:
1130 1 none
1131 1
1132 1 Returned Value:
1133 1 none
1134 1
1135 1 Side Effects:
1136 1 none
1137 1 --
1138 1 GLOBAL ROUTINE PSM$FILE_BURST ( %SBTTL 'FILE_BURST'
1139 1 SMB_CONTEXT : REF VECTOR,
1140 1 USER_CONTEXT : REF VECTOR,
1141 1 FUNCTION : REF VECTOR,
1142 1 FUNC_DESC : REF VECTOR,
1143 1 FUNC_ARG : REF VECTOR
1144 1 ) =
1145 2 BEGIN
1146 2
1147 2 LOCAL
1148 2 SCB : REF $BLOCK,
1149 2 STATUS,
1150 2 FORM_WIDTH,
1151 2 FORM_LENGTH,
1152 2 FORM_SIZE,
1153 2 PAGE_REF : REF PAGE ARRAY, ! Declare the pointer to page
1154 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
1155 2 STRING_DESC : VECTOR [2]; ! Descriptor to current string
1156 2
1157 2 SCB = .SMB_CONTEXT[0];
1158 2
1159 2 ! Check the FUNCTION requested
1160 2
1161 2 SELECTONEU .FUNCTION[0] OF
1162 2 SET
1163 2 [PSM$K_READ]:
1164 2 BEGIN
1165 2 PAGE_REF = .SCB[PSM$A_PAGE_POINTER];
1166 2
```



```
223 1167 3
224 1168
225 1169      ! Output one line at a time
226 1170      IF .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH]-6) THEN
227 1171          RETURN PSM$EOF;
228 1172
229 1173      FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
230 1174      FUNC_DESC[ADDR] = PAGE_REF[0, .SCB[PSMSL_RECORD_NUMBER],
231 1175          .SCB[PSMSL_PAGE_WIDTH]];
232 1176
233 1177      ! adjust pointer
234 1178      FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
235 1179          %CHAR(32), .FUNC_DESC[SIZE]);
236 1180
237 1181      END;
238 1182 [PSMSK_OPEN]:
239 1183     BEGIN
240 1184         GET_FORM_SIZE (.SCB);          ! Returns the WidthxLength
241 1185
242 1186         FORM_WIDTH      = .SCB[PSMSL_PAGE_WIDTH];
243 1187         FORM_LENGTH     = .SCB[PSMSL_PAGE_LENGTH];
244 1188
245 1189         RETURN_IF_ERROR_(ALLOCATE_PAGE( .SCB)); ! Get the page of memory
246 1190
247 1191         PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
248 1192
249 1193         ! Allocate the buffer for "GET_xxx" Routines
250 1194
251 1195         STRING_DESC[SIZE] = %ALLOCATION(BUFFER); ! allocate for routines
252 1196         STRING_DESC[ADDR] = BUFFER;             ! init address
253 1197
254 1198         ! No Form_feed for the burst page
255 1199
256 1200
257 1201         ! Format the page identically to File Flag
258 1202         ! Standard Burst Page 132x66: text covers rows 2 through 60,
259 1203         ! translated to frames... ref starts at 2 and length is 58.
260 1204         FILL_FILE_FLAG(
261 1205             .SCB,
262 1206             PAGE_REF[0,2,.FORM_WIDTH],
263 1207             .FORM_WIDTH,
264 1208             .FORM_LENGTH - 6 - 2 );          ! bottom margin is 6
265 1209                                             ! 2 spaces at the top
266 1210
267 1211     END;
268 1212 [PSMSK_CLOSE]:          ! Return the Page of Memory
269 1213     RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));
270 1214
271 1215 [OTHERWISE]:
272 1216     RETURN PSM$FUNNOTSUP;
273 1217
274 1218     TES; ! case .function
275 1219
276 1220 SS$_NORMAL
277 1221 END;
```

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_BURST

K 13
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 7
(4)

.TITLE SEPARATE Print Symbiont -- separation routines
.IDENT \V04-001\

.PSECT DATA,NOEXE,2

00000 BURST: .BLKB 4

.EXTRN BASSEdit, LBR\$CLOSE
.EXTRN LBR\$GET RECORD, LBR\$INI CONTROL
.EXTRN LBR\$LOOKUP KEY, LBR\$OPEN
.EXTRN LBR\$RET RMSSTV, LBR\$SET_LOCATE
.EXTRN LIB\$TRIM FILESPEC
.EXTRN LIB\$GET VM, LIB\$FREE_VM
.EXTRN STR\$ANALYZE_SDESC
.EXTRN STR\$ANALYZE_SDESC R1
.EXTRN STR\$APPEND, STR\$CONCAT
.EXTRN STR\$COPY DX, STR\$COPY R
.EXTRN STR\$FREE DX, STR\$FREE1_DX_R4
.EXTRN STR\$GET1 DX, STR\$LEFT
.EXTRN STR\$PREFIX, STR\$RIGHT
.EXTRN PSMS\$ HANGUP DISPATCH ENTRY
.EXTRN PSMS\$ BUFFEROVF, PSMS\$ EOF
.EXTRN PSMS\$ ESCAPE, PSMS\$ FLUSH
.EXTRN PSMS\$ FUNNOTSUP, PSMS\$ INVITMCO
.EXTRN PSMS\$ INVVMSC, PSMS\$ MODNOTFND
.EXTRN PSMS\$ NEWPAGE, PSMS\$ NOFILEID
.EXTRN PSMS\$ OSCTOOLON, PSMS\$ PENDING
.EXTRN PSMS\$ SUSPEND, PSMS\$ TOOMANYLEV
.EXTRN SMB\$ INVSTMNR, SMB\$ INVSTRLEV
.EXTRN SMB\$ NOMOREITEMS
.EXTRN PSMS\$ BANNER, PSMS\$ READ_ITEM_DX

.PSECT CODE, NOWRT, 2

				003C 00000	.ENTRY PSMS\$FILE_BURST, Save R2,R3,R4,R5	: 1138
		5E	FDF8	CE 9E 00002	MOVAB -520(SP), SP	: 1158
		52	04	BC D0 00007	MOVL @SMB_CONTEXT, SCB	: 1162
		50	0C	BC D0 0000B	MOVL @FUNCTION, R0	: 1164
		05		50 D1 0000F	CMPL R0, #5	: 1166
				41 12 00012	BNEQ 2\$: 1168
		55	01FC	C2 D0 00014	MOVL 508(SCB), PAGE_REF	: 1169
50	01F8	C2		06 C3 00019	SUBL3 #6, 504(SCB), R0	: 1171
		50	026C	C2 D1 0001F	CMPL 620(SCB), R0	: 1173
				08 15 00024	BLEQ 1\$: 1177
		50	00000000G	8F D0 00026	MOVL #PSMS_EOF, R0	: 1176
				04 0002D	RET	
		53	10	AC D0 0002E 1\$:	MOVL FUNC_DESC, R3	: 1162
		63	0200	C2 D0 00032	MOVL 512(SCB), (R3)	: 1181
04	50	026C	0200	C2 C5 00037	MULL3 512(SCB), 620(SCB), R0	
		50		55 C1 0003F	ADDL3 PAGE_REF, R0, 4(R3)	
				63 DD 00044	PUSHL (R3)	
				20 DD 00046	PUSHL #32	
			04	A3 DD 00048	PUSHL 4(R3)	
		0000V	CF	03 FB 0004B	CALLS #3, DELIMIT_STRING_NOT	
		63		50 D0 00050	MOVL R0, (R3)	
				58 11 00053	BRB 5\$	
		04		50 D1 00055 2\$:	CMPL R0, #4	

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_BURST

L 13
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 8
(4)

0000V	CF		3B	12	00058	BNEQ	3\$:	
	53		52	DD	0005A	PUSHL	SCB	:	1184
	54	0200	01	FB	0005C	CALLS	#1, GET_FORM_SIZE	:	
		01F8	C2	D0	00061	MOVL	512(SCB), FORM_WIDTH	:	1187
			C2	D0	00066	MOVL	504(SCB), FORM_LENGTH	:	1188
0000V	CF		52	DD	0006B	PUSHL	SCB	:	1190
	3B		01	FB	0006D	CALLS	#1, ALLOCATE_PAGE	:	
	55	01FC	50	E9	00072	BLBC	STATUS, 6\$:	
	6E	0200	C2	D0	00075	MOVL	508(SCB), PAGE_REF	:	1192
04	AE	08	8F	3C	0007A	MOVZWL	#512, STRING_DESC	:	1196
		F8	AE	9E	0007F	MOVAB	BUFFER, STRING_DESC+4	:	1197
			A4	9F	00084	PUSHAB	-8(FORM_LENGTH)	:	1208
			53	DD	00087	PUSHL	FORM_WIDTH	:	1207
			6543	3F	00089	PUSHAW	(PAGE_REF)[FORM_WIDTH]	:	1206
			52	DD	0008C	PUSHL	SCB	:	
0000V	CF		04	FB	0008E	CALLS	#4, FILL_FILE_FLAG	:	
	02		18	11	00093	BRB	5\$:	1162
			50	D1	00095	CMPL	R0, #2	:	1211
			0B	12	00098	BNEQ	4\$:	
0000V	CF		52	DD	0009A	PUSHL	SCB	:	1212
	09		01	FB	0009C	CALLS	#1, DEALLOCATE_PAGE	:	
			50	E8	000A1	BLBS	STATUS, 5\$:	
				04	000A4	RET		:	
	50	00000000G	8F	D0	000A5	MOVL	#PSMS_FUNNOTSUP, R0	:	1215
				04	000AC	RET		:	
	50		01	D0	000AD	MOVL	#1, R0	:	1221
			04	000B0	6\$:	RET		:	

; Routine Size: 177 bytes, Routine Base: CODE + 0000


```
279 1222 1 %sbttl 'PSMSFILE_FLAG - Print a File Flag Page'
280 1223 1 Functional Description:
281 1224 1 This routine controls the creation of the file flag page. The
282 1225 1 FUNCTION code dictates the action taken in creation.
283 1226 1 FUNCTION:
284 1227 1 OPEN - Allocate and create the file Flag Page
285 1228 1 READ - Return the current line of the file Flag Page
286 1229 1 CLOSE - Return the buffer allocated on OPEN
287 1230 1
288 1231 1 Formal Parameters:
289 1232 1 SMB_CONTEXT - Pointer to the SMB
290 1233 1 USER_CONTEXT - User defined pointer (not used here)
291 1234 1 FUNCTION - OPEN, READ, CLOSE
292 1235 1 FUNC_DESC - Pointer to functionally dependent descriptor
293 1236 1 FUNC_ARG - Pointer to functionally dependent argument
294 1237 1
295 1238 1 Implicit Inputs:
296 1239 1 none
297 1240 1
298 1241 1 Implicit Outputs:
299 1242 1 none
300 1243 1
301 1244 1 Returned Value:
302 1245 1 none
303 1246 1
304 1247 1 Side Effects:
305 1248 1 none
306 1249 1 --
307 1250 1
308 1251 1 GLOBAL ROUTINE PSMSFILE_FLAG ( %SBTTL 'FILE_FLAG'
309 1252 1 SMB_CONTEXT : REF VECTOR,
310 1253 1 USER_CONTEXT : REF VECTOR,
311 1254 1 FUNCTION : REF VECTOR,
312 1255 1 FUNC_DESC : REF VECTOR,
313 1256 1 FUNC_ARG : REF VECTOR
314 1257 1 ) =
315 1258 2 BEGIN
316 1259 2
317 1260 2 LOCAL
318 1261 2 SCB : REF $BLOCK,
319 1262 2 STATUS,
320 1263 2 FORM_WIDTH,
321 1264 2 FORM_LENGTH,
322 1265 2 FORM_SIZE,
323 1266 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
324 1267 2 STRING_DESC : VECTOR [2], ! Descriptor to current string
325 1268 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
326 1269 2 to page
327 1270 2
328 1271 2 SCB = .SMB_CONTEXT[0];
329 1272 2
330 1273 2 ! Check the FUNCTION requested
331 1274 2
332 1275 2 SELECTONEU .FUNCTION[0] OF
333 1276 2 SET
334 1277 2 [PSMSK_READ]:
335 1278 3 BEGIN
```

```
336 1279 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
337 1280
338 1281 IF ( .SCB[PSMSL_RECORD_NUMBER] GTR .SCB[PSMSL_PAGE_LENGTH])
339 1282     OR
340 1283     ( NOT .SEPARATE_FLAG (FILE BURST) AND
341 1284     .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 6) )
342 1285 THEN
343 1286     RETURN PSMS_EOF;
344 1287
345 1288 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
346 1289 FUNC_DESC[ADDR] = PAGE_REF[0, .SCB[PSMSL_RECORD_NUMBER],
347 1290     .SCB[PSMSL_PAGE_WIDTH]];
348 1291
349 1292     ! adjust pointer
350 1293     FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
351 1294     XCHAR(32), .FUNC_DESC[SIZE]);
352 1295
353 1296 END;
354 1297
355 1298 [PSMSK_OPEN]:
356 1299 BEGIN
357 1300
358 1301 GET_FORM_SIZE (.SCB);
359 1302
360 1303
361 1304 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
362 1305 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
363 1306
364 1307 RETURN_IF_ERROR(ALLOCATE_PAGE(.SCB));
365 1308
366 1309 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
367 1310
368 1311 ! Always start at top of page
369 1312
370 1313 PAGE_REF[0,0,.FORM_WIDTH] = PSMSK_CHAR_FF;
371 1314
372 1315 ! Standard Flag Page 132x66: text covers rows 1 through 58,
373 1316 ! translated to frames... ref starts at 1 and length is 57.
374 1317
375 1318 FILL_FILE_FLAG(.SCB,
376 1319     PAGE_REF[0,1,.FORM_WIDTH],
377 1320     .FORM_WIDTH,
378 1321     .FORM_LENGTH - 6 - 2 - 1);
379 1322
380 1323 ! Burst always appears on the flag page, starting at page length - 5
381 1324 ! and continuing to page length. This leaves a two blank lines
382 1325 ! between file flag footer bar and file burst header bar.
383 1326 ! This IS the right way to perform a BURST over the crease !
384 1327
385 1328 IF (.SEPARATE_FLAG(FILE_BURST)) THEN
386 1329 BEGIN
387 1330     STRING_DESC[SIZE] = XALLOCATION(BUFFER);
388 1331     STRING_DESC[ADDR] = BUFFER;
389 1332
390 1333     ! determine the correct size of the string to insert
391 1334
392 1335     GET_VMS_LOGO
```

```
393 1336 4      (.SCB,  
394 1337 4      STRING_DESC[0],          ! Buffer descriptor  
395 1338 4      STRING_DESC[SIZE]);      ! Returned length  
396 1339 4  
397 1340 4      INSERT_FRAME  
398 1341 4      (.SCB,  
399 1342 4      STRING_DESC[0],  
400 1343 4      PAGE_REF[10,.FORM_LENGTH-5,.FORM_WIDTH],  
401 1344 4      .FORM_WIDTH-20, 1);  
402 1345 4      INSERT_FRAME  
403 1346 4      (.SCB,  
404 1347 4      STRING_DESC[0],  
405 1348 4      PAGE_REF[14,.FORM_LENGTH-4,.FORM_WIDTH],  
406 1349 4      .FORM_WIDTH-16, 1);  
407 1350 4      INSERT_FRAME  
408 1351 4      (.SCB,  
409 1352 4      STRING_DESC[0],  
410 1353 4      PAGE_REF[10,.FORM_LENGTH-3,.FORM_WIDTH],  
411 1354 4      .FORM_WIDTH-20, 1);  
412 1355 4      INSERT_FRAME  
413 1356 4      (.SCB,  
414 1357 4      STRING_DESC[0],  
415 1358 4      PAGE_REF[14,.FORM_LENGTH-2,.FORM_WIDTH],  
416 1359 4      .FORM_WIDTH-16, 1);  
417 1360 4      INSERT_FRAME  
418 1361 4      (.SCB,  
419 1362 4      STRING_DESC[0],  
420 1363 4      PAGE_REF[10,.FORM_LENGTH-1,.FORM_WIDTH],  
421 1364 4      .FORM_WIDTH-20, 1);  
422 1365 4      INSERT_FRAME  
423 1366 4      (.SCB,  
424 1367 4      STRING_DESC[0],  
425 1368 4      PAGE_REF[14,.FORM_LENGTH,.FORM_WIDTH],  
426 1369 4      .FORM_WIDTH-16, 1);  
427 1370 3      END;  
428 1371 2      END;  
429 1372 2  
430 1373 2      [PSMSK CLOSE]:          ! Return the Page of Memory  
431 1374 2      RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));  
432 1375 2  
433 1376 2      [OTHERWISE]:  
434 1377 2      RETURN PSM$_FUNNOTSUP;  
435 1378 2  
436 1379 2      TES; ! case .function  
437 1380 2  
438 1381 2      SSS_NORMAL  
439 1382 2  
440 1383 1      END;
```

```
56 0000V 007C 00000  
5E FDF8 CE 9E 00002  
54 04 BC D0 0000C
```

```
.ENTRY PSM$FILE_FLAG, Save R2,R3,R4,R5,R6  
MOVAB INSERT_FRAME, R6  
MOVAB -520(SP), SP  
MOVL @SMB_CONTEXT, SCB
```

```
: 1251  
:  
: 1271
```


	50	0C	BC	D0	00010	MOVL	3FUNCTION, R0	1275
	05		50	D1	00014	CMPL	R0, #5	1277
	53	01FC	4E	12	00017	BNEQ	4\$	
	50	026C	C4	D0	00019	MOVL	508(SCB), PAGE_REF	1279
01F8	C4		C4	D0	0001E	MOVL	620(SCB), R0	1281
			50	D1	00023	CMPL	R0, 504(SCB)	
			10	14	00028	BGTR	1\$	
51	01F8	0154	C4	E8	0002A	BLBS	340(SCB), 2\$	1283
			06	C3	0002F	SUBL3	#6, 504(SCB), R1	1284
			50	D1	00035	CMPL	R0, R1	
			08	15	00038	BLEQ	2\$	
	50	00000000G	8F	D0	0003A	1\$: MOVL	#PSMS_EOF, R0	1286
				04	00041	RET		
	52	10	AC	D0	00042	2\$: MOVL	FUNC_DESC, R2	1288
	62	0200	C4	D0	00046	MOVL	512(SCB), (R2)	
	50	0200	C4	C4	0004B	MULL2	512(SCB), R0	1290
04	A2		53	C1	00050	ADDL3	PAGE_REF, R0, 4(R2)	
			62	DD	00055	PUSHL	(R2)	1294
			20	DD	00057	PUSHL	#32	1293
		04	A2	DD	00059	PUSHL	4(R2)	
0000V	CF		03	FB	0005C	CALLS	#3, DELIMIT_STRING_NOT	
	62		50	D0	00061	MOVL	R0, (R2)	
	04		00FC	31	00064	3\$: BRW	9\$	1275
			50	D1	00067	4\$: CMPL	R0, #4	1298
			03	13	0006A	BEQL	5\$	
			00DC	31	0006C	BRW	7\$	
			54	DD	0006F	5\$: PUSHL	SCB	1301
0000V	CF		01	FB	00071	CALLS	#1, GET_FORM_SIZE	
	55	0200	C4	D0	00076	MOVL	512(SCB), FORM_WIDTH	1304
	52	01F8	C4	D0	0007B	MOVL	504(SCB), FORM_LENGTH	1305
			54	DD	00080	PUSHL	SCB	1307
0000V	CF		01	FB	00082	CALLS	#1, ALLOCATE_PAGE	
	01		50	E8	00087	BLBS	STATUS, 6\$	
				04	0008A	RET		
	53	01FC	C4	D0	0008B	6\$: MOVL	508(SCB), PAGE_REF	1309
	63		0C	90	00090	MOVB	#12, (PAGE_REF)	1313
		F7	A2	9F	00093	PUSHAB	-9(FORM_LENGTH)	1320
			55	DD	00096	PUSHL	FORM_WIDTH	1319
			6543	9F	00098	PUSHAB	(FORM_WIDTH)[PAGE_REF]	1318
			54	DD	0009B	PUSHL	SCB	
0000V	CF		04	FB	0009D	CALLS	#4, FILL_FILE_FLAG	
	BD	0154	C4	E9	000A2	BLBC	340(SCB), 3\$	1328
	6E	0200	8F	3C	000A7	MOVZWL	#512, STRING_DESC	1330
04	AE	08	AE	9E	000AC	MOVAB	BUFFER, STRING_DESC+4	1331
			5E	DD	000B1	PUSHL	SP	1338
		04	AE	9F	000B3	PUSHAB	STRING_DESC	1337
			54	DD	000B6	PUSHL	SCB	1336
0000V	CF		03	FB	000B8	CALLS	#3, GET_VMS_LOGO	
			01	DD	000BD	PUSHL	#1	1343
		EC	A5	9F	000BF	PUSHAB	-20(FORM_WIDTH)	1344
	50	FB	A2	9E	000C2	MOVAB	-5(R2), R0	1343
	50		55	C4	000C6	MULL2	FORM_WIDTH, R0	
		0A	A043	9F	000C9	PUSHAB	10(R0)[PAGE_REF]	
		0C	AE	9F	000CD	PUSHAB	STRING_DESC	1342
			54	DD	000D0	PUSHL	SCB	1343
	66		05	FB	000D2	CALLS	#5, INSERT_FRAME	
			01	DD	000D5	PUSHL	#1	1348

	FO	A5	9F	000D7	PUSHAB	-16(FORM WIDTH)	1349
50	FC	A2	9E	000DA	MOVAB	-4(R2), R0	1348
50		55	C4	000DE	MULL2	FORM WIDTH, R0	
	OE	A043	9F	000E1	PUSHAB	14(R0)[PAGE_REF]	
	OC	AE	9F	000E5	PUSHAB	STRING_DESC	1347
		54	DD	000E8	PUSHL	SCB	1348
66		05	FB	000EA	CALLS	#5, INSERT_FRAME	
		01	DD	000ED	PUSHL	#1	1353
	EC	A5	9F	000EF	PUSHAB	-20(FORM WIDTH)	1354
50	FD	A2	9E	000F2	MOVAB	-3(R2), R0	1353
50		55	C4	000F6	MULL2	FORM WIDTH, R0	
	OA	A043	9F	000F9	PUSHAB	10(R0)[PAGE_REF]	
	OC	AE	9F	000FD	PUSHAB	STRING_DESC	1352
		54	DD	00100	PUSHL	SCB	1353
66		05	FB	00102	CALLS	#5, INSERT_FRAME	
		01	DD	00105	PUSHL	#1	1358
	FO	A5	9F	00107	PUSHAB	-16(FORM WIDTH)	1359
50	FE	A2	9E	0010A	MOVAB	-2(R2), R0	1358
50		55	C4	0010E	MULL2	FORM WIDTH, R0	
	OE	A043	9F	00111	PUSHAB	14(R0)[PAGE_REF]	
	OC	AE	9F	00115	PUSHAB	STRING_DESC	1357
		54	DD	00118	PUSHL	SCB	1358
66		05	FB	0011A	CALLS	#5, INSERT_FRAME	
		01	DD	0011D	PUSHL	#1	1363
	EC	A5	9F	0011F	PUSHAB	-20(FORM WIDTH)	1364
50	FF	A2	9E	00122	MOVAB	-1(R2), R0	1363
50		55	C4	00126	MULL2	FORM WIDTH, R0	
	OA	A043	9F	00129	PUSHAB	10(R0)[PAGE_REF]	
	OC	AE	9F	0012D	PUSHAB	STRING_DESC	1362
		54	DD	00130	PUSHL	SCB	1363
66		05	FB	00132	CALLS	#5, INSERT_FRAME	
		01	DD	00135	PUSHL	#1	1368
	FO	A5	9F	00137	PUSHAB	-16(FORM WIDTH)	1369
52		55	C4	0013A	MULL2	FORM WIDTH, R2	1368
	OE	A243	9F	0013D	PUSHAB	14(R2)[PAGE_REF]	
	OC	AE	9F	00141	PUSHAB	STRING_DESC	1367
		54	DD	00144	PUSHL	SCB	1368
66		05	FB	00146	CALLS	#5, INSERT_FRAME	
		18	11	00149	BRB	9\$	1275
02		50	D1	0014B	CMPL	R0, #2	1373
		0B	12	0014E	BNEQ	8\$	
		54	DD	00150	PUSHL	SCB	1374
0000V	CF	01	FB	00152	CALLS	#1, DEALLOCATE_PAGE	
09		50	FB	00157	BLBS	STATUS, 9\$	
			14	0015A	RET		
50	00000000G	8F		0015B	MOVL	#PSM\$_FUNNOTSUP, R0	1377
			04	00162	RET		
50		01	DD	00163	MOVL	#1, R0	1383
			04	00166	RET		

; Routine Size: 359 bytes, Routine Base: CODE + 00B1

```

442 1384 1 %sbttl 'PSMSFILE_TRAILER - Print a File Trailer Page'
443 1385 1 Functional Description:
444 1386 1 This routine controls the creation of the file flag page. The
445 1387 1 FUNCTION code dictates the action taken in creation.
446 1388 1 FUNCTION:
447 1389 1 OPEN - Allocate and create the file Flag Page
448 1390 1 READ - Return the current line of the file Flag Page
449 1391 1 CLOSE - Return the buffer allocated on OPEN
450 1392 1
451 1393 1 Formal Parameters:
452 1394 1 SMB_CONTEXT - Pointer to the SMB
453 1395 1 USER_CONTEXT - User defined pointer (not used here)
454 1396 1 FUNCTION - OPEN, READ, CLOSE
455 1397 1 FUNC_DESC - Pointer to functionally dependent descriptor
456 1398 1 FUNC_ARG - Pointer to functionally dependent argument
457 1399 1
458 1400 1 Implicit Inputs:
459 1401 1 none
460 1402 1
461 1403 1 Implicit Outputs:
462 1404 1 none
463 1405 1
464 1406 1 Returned Value:
465 1407 1 none
466 1408 1
467 1409 1 Side Effects:
468 1410 1 none
469 1411 1 --
470 1412 1
471 1413 1 GLOBAL ROUTINE PSMSFILE_TRAILER ( %SBTTL 'FILE_TRAILER'
472 1414 1 SMB_CONTEXT : REF VECTOR,
473 1415 1 USER_CONTEXT : REF VECTOR,
474 1416 1 FUNCTION : REF VECTOR,
475 1417 1 FUNC_DESC : REF VECTOR,
476 1418 1 FUNC_ARG : REF VECTOR
477 1419 1 ) =
478 1420 2 BEGIN
479 1421 2
480 1422 2 LITERAL
481 1423 2 TRAILING = 1;
482 1424 2 LOCAL
483 1425 2 SCB : REF $BLOCK,
484 1426 2 STATUS,
485 1427 2 FORM_WIDTH,
486 1428 2 FORM_LENGTH,
487 1429 2 FORM_SIZE,
488 1430 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
489 1431 2 ! to page
490 1432 2
491 1433 2 SCB = .SMB_CONTEXT[0];
492 1434 2
493 1435 2 ! (check the FUNCTION requested
494 1436 2
495 1437 2 SELECTONEU .FUNCTION[0] OF
496 1438 2 SET
497 1439 2 [PSMSK READ]:
498 1440 2 BEGIN
```



```
499 1441 3 LOCAL TEMP_PTR;
500 1442
501 1443 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
502 1444
503 1445 ! Output one line at a time
504 1446 IF .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 6 - 2) THEN
505 1447 RETURN PSMS_EOF;
506 1448
507 1449 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
508 1450 FUNC_DESC[ADDR] = PAGE_REF[0, .SCB[PSMSL_RECORD_NUMBER],
509 1451 .SCB[PSMSL_PAGE_WIDTH]];
510 1452
511 1453 ! adjust pointer
512 1454 FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
513 1455 %CHAR(32), .FUNC_DESC[SIZE]);
514 1456
515 1457 END;
516 1458 [PSMSK_OPEN]:
517 1459 BEGIN
518 1460
519 1461 GET_FORM_SIZE (.SCB); ! Returns the WidthxLength
520 1462
521 1463
522 1464 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
523 1465 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
524 1466
525 1467 RETURN_IF_ERROR(ALLOCATE_PAGE(.SCB)); ! Get the page of memory
526 1468
527 1469 PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
528 1470
529 1471 ! Always start at top of page
530 1472
531 1473 PAGE_REF[0,0,.FORM_WIDTH] = PSMSK_CHAR_FF; ! form feed in 0 pos.
532 1474
533 1475 ! Standard Trailer Page 132x66: text covers rows 1 through 58,
534 1476 ! translated to frames... ref starts at 1 and length is 57.
535 1477
536 1478 FILL_FILE_TRAILER(.SCB,
537 1479 PAGE_REF[0,1,.FORM_WIDTH],
538 1480 .FORM_WIDTH,
539 1481 .FORM_LENGTH - 6 - 2 - 1); ! total form length...
540 1482 ! ..6 burst, 2 spaces
541 1483 ! ...top margin of 1
542 1484
543 1485 [PSMSK_CLOSE]: ! Return the Page of Memory
544 1486 RETURN_IF_ERROR(DEALLOCATE_PAGE(.SCB));
545 1487
546 1488 [OTHERWISE]:
547 1489 RETURN PSMS_FUNNOTSUP;
548 1490
549 1491 TES; ! case .function
550 1492
551 1493 $$$_NORMAL
552 1494 END;
```

				003C	00000	.ENTRY	PSMS\$FILE TRAILER, Save R2,R3,R4,R5	1413
	52	04	BC	D0	00002	MOVL	@SMB_CONTEXT, SCB	1433
	50	0C	BC	D0	00006	MOVL	@FUNCTION, R0	1437
	05		50	D1	0000A	CMPL	R0, #5	1439
			41	12	0000D	BNEQ	2\$	
	54	01FC	C2	D0	0000F	MOVL	508(SCB), PAGE_REF	1443
50	01F8		08	C3	00014	SUBL3	#8, 504(SCB), R0	1445
	50	026C	C2	D1	0001A	CMPL	620(SCB), R0	
			08	15	0001F	BLEQ	1\$	
	50	00000000G	8F	D0	00021	MOVL	#PSMS_EOF, R0	1446
				04	00028	RET		
	53	10	AC	D0	00029	1\$: MOVL	FUNC_DESC, R3	1448
	63	0200	C2	D0	0002D	MOVL	512(SCB), (R3)	
04	50	026C	C2	C5	00032	MULL3	512(SCB), 620(SCB), R0	1450
A3	50		54	C1	0003A	ADDL3	PAGE_REF, R0, 4(R3)	
			63	DD	0003F	PJSHL	(R3)	1454
			20	DD	00041	PUSHL	#32	1453
		04	A3	DD	00043	PUSHL	4(R3)	
0000V	CF		03	FB	00046	CALLS	#3, DELIMIT_STRING_NOT	
	63		50	D0	00048	MOVL	R0, (R3)	
	04		51	11	0004E	BRB	5\$	1437
			50	D1	00050	2\$: CMPL	R0, #4	1458
			34	12	00053	BNEQ	3\$	
			52	DD	00055	PUSHL	SCB	1461
0000V	CF		01	FB	00057	CALLS	#1, GET_FORM_SIZE	
	55	0200	C2	D0	0005C	MOVL	512(SCB), FORM_WIDTH	1464
	53	01F8	C2	D0	00061	MOVL	504(SCB), FORM_LENGTH	1465
			52	DD	00066	PUSHL	SCB	1467
0000V	CF		01	FB	00068	CALLS	#1, ALLOCATE_PAGE	
	34		50	E9	0006D	BLBC	STATUS, 6\$	
	54	01FC	C2	D0	00070	MOVL	508(SCB), PAGE_REF	1469
	64		0C	90	00075	MOVB	#12, (PAGE_REF)	1473
		F7	A3	9F	00078	PUSHAB	-9(FORM_LENGTH)	1480
			55	DD	0007B	PUSHL	FORM_WIDTH	1479
			6544	9F	0007D	PUSHAB	(FORM_WIDTH)[PAGE_REF]	1478
			52	DD	00080	PUSHL	SCB	
0000V	CF		04	FB	00082	CALLS	#4, FILL_FILE_TRAILER	
	02		18	11	00087	BRB	5\$	1437
			50	D1	00089	3\$: CMPL	R0, #2	1484
			08	12	0008C	BNEQ	4\$	
			52	DD	0008E	PUSHL	SCB	1485
0000V	CF		01	FB	00090	CALLS	#1, DEALLOCATE_PAGE	
	09		50	E8	00095	BLBS	STATUS, 5\$	
			04	00098	RET			
	50	00000000G	8F	D0	00099	4\$: MOVL	#PSMS_FUNNOTSUP, R0	1488
			04	000A0	RET			
	50		01	D0	000A1	5\$: MOVL	#1, R0	1494
			04	000A4	6\$: RET			

; Routine Size: 165 bytes, Routine Base: CODE + 0218

```
1495 1 %sbttl 'PSMSJOB_BURST - Print a Job Burst Page'
1496 1 Functional Description:
1497 1 This routine controls the creation of the job burst page. The
1498 1 FUNCTION code dictates the action taken in creation.
1499 1 FUNCTION:
1500 1 OPEN - Allocate and create the Job Burst Page
1501 1 READ - Return the current line of the Job Burst Page
1502 1 CLOSE - Return the buffer allocated on OPEN
1503 1
1504 1 Formal Parameters:
1505 1 SMB_CONTEXT - Pointer to the SMB
1506 1 USER_CONTEXT - User defined pointer (not used here)
1507 1 FUNCTION - OPEN, READ, CLOSE
1508 1 FUNC_DESC - Pointer to functionally dependent descriptor
1509 1 FUNC_ARG - Pointer to functionally dependent argument
1510 1
1511 1 Implicit Inputs:
1512 1 none
1513 1
1514 1 Implicit Outputs:
1515 1 none
1516 1
1517 1 Returned Value:
1518 1 none
1519 1
1520 1 Side Effects:
1521 1 none
1522 1 --
1523 1 GLOBAL ROUTINE PSMSJOB_BURST ( %SBTTL 'JOB_BURST'
1524 1 SMB_CONTEXT : REF VECTOR,
1525 1 USER_CONTEXT : REF VECTOR,
1526 1 FUNCTION : REF VECTOR,
1527 1 FUNC_DESC : REF VECTOR,
1528 1 FUNC_ARG : REF VECTOR
1529 1 ) =
1530 2 BEGIN
1531 2
1532 2 LOCAL
1533 2 SCB : REF $BBLOCK,
1534 2 STATUS,
1535 2 FORM_WIDTH,
1536 2 FORM_LENGTH,
1537 2 FORM_SIZE,
1538 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
1539 2 ! to page
1540 2
1541 2 SCB = .SMB_CONTEXT[0];
1542 2
1543 2 ! Check the FUNCTION requested
1544 2
1545 2 SELECTONEU .FUNCTION[0] OF
1546 2 SET
1547 2 [PSMSK_READ]:
1548 2 BEGIN
1549 2 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
1550 2
1551 2 IF .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 6)
! Output one line at a time
```

```
611 1552 THEN
612 1553     RETURN PSMS_EOF;
613 1554
614 1555     FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
615 1556     FUNC_DESC[ADDR] = PAGE_REF[0, .SCB[PSMSL_RECORD_NUMBER],
616 1557                     .SCB[PSMSL_PAGE_WIDTH]];
617 1558
618 1559     ! adjust pointer
619 1560     FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
620 1561                                         %CHAR(32), .FUNC_DESC[SIZE]);
621 1562
622 1563     END;
623 1564
624 1565 [PSMSK_OPEN]:
625 1566     BEGIN
626 1567
627 1568     GET_FORM_SIZE (.SCB);                ! Returns the WidthxLength
628 1569
629 1570
630 1571     FORM_WIDTH      = .SCB[PSMSL_PAGE_WIDTH];
631 1572     FORM_LENGTH     = .SCB[PSMSL_PAGE_LENGTH];
632 1573
633 1574     RETURN_IF_ERROR(ALLOCATE_PAGE(.SCB)); ! Get the page of memory
634 1575
635 1576     PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
636 1577
637 1578     ! No form_feed on a burst page
638 1579
639 1580     ! Standard Burst Page 132x66: text covers rows 2 through 60,
640 1581     ! translated to frames... ref starts at 2 and length is 58.
641 1582     FILL_JOB_FLAG( .SCB,
642 1583                  PAGE_REF[0, 2, .FORM_WIDTH],
643 1584                  .FORM_WIDTH,
644 1585                  .FORM_LENGTH - 6 - 2 );    ! 6 blank lines
645 1586                                           ! top margin is 2
646 1587
647 1588     END;
648 1589 [PSMSK_CLOSE]:                ! Return the Page of Memory
649 1590     RETURN_IF_ERROR(DEALLOCATE_PAGE(.SCB));
650 1591
651 1592 [OTHERWISE]:
652 1593     RETURN PSMS_FUNNOTSUP;
653 1594
654 1595     TES; ! case .function
655 1596
656 1597     SSS_NORMAL
657 1598     END;
```

```
52      04      003C 00000
50      0C      BC  D0 00002
05      50      BC  D0 00006
          50      D1 0000A
          41      12 0000D
```

```
.ENTRY PSMSJOB BURST, Save R2,R3,R4,R5
MOVL   @SMB_CONTEXT, SCB
MOVL   @FUNCTION, R0
CML    R0, #5
BNEQ   2$
```

```
1523
1541
1545
1547
```


SEPARATE
V04-001

Print Symbiont -- separation routines
JOB_BURST

J 14
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 19
(7)

		55	01FC	C2	D0	0000F	MOVL	508(SCB), PAGE_REF	1549
50	01F8	C2		06	C3	00014	SUBL3	#6, 504(SCB), R0	1551
		50	026C	C2	D1	0001A	CMPL	620(SCB), R0	
				08	15	0001F	BLEQ	1\$	
		50	00000000G	8F	D0	00021	MOVL	#PSMS_EOF, R0	1553
					04	00028	RET		
		53	10	AC	D0	00029	1\$: MOVL	FUNC_DESC, R3	1555
		63	0200	C2	D0	0002D	MOVL	512(SCB), (R3)	
04	50	026C	0200	C2	C5	00032	MJLL3	512(SCB), 620(SCB), R0	1557
A3				55	C1	0003A	ADDL3	PAGE_REF, R0, 4(R3)	
				63	DD	0003F	PUSHL	(R3)	1561
				20	DD	00041	PUSHL	#32	1560
			04	A3	DD	00043	PUSHL	4(R3)	
	0000V	CF		03	FB	00046	CALLS	#3, DELIMIT_STRING_NOT	
		63		50	D0	0004B	MOVL	R0, (R3)	
				4E	11	0004E	BRB	5\$	1545
		04		50	D1	00050	2\$: CMPL	R0, #4	1565
				31	12	00053	BNEQ	3\$	
				52	DD	00055	PUSHL	SCB	1568
	0000V	CF		01	FB	00057	CALLS	#1, GET_FORM_SIZE	
		53	0200	C2	D0	0005C	MOVL	512(SCB), FORM_WIDTH	1571
		54	01F8	C2	D0	00061	MOVL	504(SCB), FORM_LENGTH	1572
				52	DD	00066	PUSHL	SCB	1574
	0000V	CF		01	FB	00068	CALLS	#1, ALLOCATE_PAGE	
		31		50	E9	0006D	BLBC	STATUS, 6\$	
		55	01FC	C2	D0	00070	MOVL	508(SCB), PAGE_REF	1576
			F8	A4	9F	00075	PUSHAB	-8(FORM_LENGTH)	1585
				53	DD	00078	PUSHL	FORM_WIDTH	1584
				6543	3F	0007A	PUSHAW	(PAGE_REF)[FORM_WIDTH]	1583
				52	DD	0007D	PUSHL	SCB	
	0000V	CF		04	FB	0007F	CALLS	#4, FILL_JOB_FLAG	
				18	11	00084	BRB	5\$	1545
		02		50	D1	00086	3\$: CMPL	R0, #2	1588
				0B	12	00089	BNEQ	4\$	
				52	DD	0008B	PUSHL	SCB	1589
	0000V	CF		01	FB	0008D	CALLS	#1, DEALLOCATE_PAGE	
		09		50	E8	00092	BLBS	STATUS, 5\$	
				04		00095	RET		
		50	00000000G	8F	D0	00096	4\$: MOVL	#PSMS_FUNNOTSUP, R0	1592
					04	0009D	RET		
		50		01	D0	0009E	5\$: MOVL	#1, R0	1598
					04	000A1	6\$: RET		

; Routine Size: 162 bytes, Routine Base: CODE + 02B0

```
659 1599 1 %sbttl 'PSMSJOB_FLAG - Print a Job Flag Page'
660 1600 1 Functional Description:
661 1601 1 This routine controls the creation of the job flag page. The
662 1602 1 FUNCTION code dictates the action taken in creation.
663 1603 1 FUNCTION:
664 1604 1 OPEN - Allocate and create the Job Flag Page
665 1605 1 READ - Return the current line of the Job Flag Page
666 1606 1 CLOSE - Return the buffer allocated on OPEN
667 1607 1
668 1608 1 Formal Parameters:
669 1609 1 SMB_CONTEXT - Pointer to the SMB
670 1610 1 USER_CONTEXT - User defined pointer (not used here)
671 1611 1 FUNCTION - OPEN, READ, CLOSE
672 1612 1 FUNC_DESC - Pointer to functionally dependent descriptor
673 1613 1 FUNC_ARG - Pointer to functionally dependent argument
674 1614 1
675 1615 1 Implicit Inputs:
676 1616 1 none
677 1617 1
678 1618 1 Implicit Outputs:
679 1619 1 none
680 1620 1
681 1621 1 Returned Value:
682 1622 1 none
683 1623 1
684 1624 1 Side Effects:
685 1625 1 none
686 1626 1 --
687 1627 1 GLOBAL ROUTINE PSMSJOB_FLAG ( %SBTTL 'JOB_FLAG'
688 1628 1 SMB_CONTEXT : REF VECTOR,
689 1629 1 USER_CONTEXT : REF VECTOR,
690 1630 1 FUNCTION : REF VECTOR,
691 1631 1 FUNC_DESC : REF VECTOR,
692 1632 1 FUNC_ARG : REF VECTOR
693 1633 1 ) =
694 1634 2 BEGIN
695 1635 2 LITERAL
696 1636 2 TRAILING = 1;
697 1637 2
698 1638 2 LOCAL
699 1639 2 SCB : REF $BLOCK,
700 1640 2 STATUS,
701 1641 2 FORM_WIDTH,
702 1642 2 FORM_LENGTH,
703 1643 2 FORM_SIZE,
704 1644 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
705 1645 2 STRING_DESC : VECTOR [2], ! Descriptor to current string
706 1646 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
707 1647 2 ! to page
708 1648 2
709 1649 2 SCB = .SMB_CONTEXT[0];
710 1650 2
711 1651 2 ! Check the FUNCTION requested
712 1652 2
713 1653 2 SELECTONEU .FUNCTION[0] OF
714 1654 2 SET
715 1655 2 [PSMSK_READ];
```

```
716 1656 3 BEGIN
717 1657 3 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
718 1658 3 ! Output one line at a time
719 1659 4 IF ( .SCB[PSMSL_RECORD_NUMBER] GTR .SCB[PSMSL_PAGE_LENGTH])
720 1660 3 OR
721 1661 4 ( NOT .SEPARATE_FLAG (JOB BURST) AND .SCB[PSMSL_RECORD_NUMBER]
722 1662 4 GEQ (.SCB[PSMSL_PAGE_LENGTH] - 6 ) )
723 1663 3 THEN
724 1664 3 RETURN PSMS_EOF;
725 1665 3
726 1666 3 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
727 1667 3 FUNC_DESC[ADDR] = PAGE_REF[0, .SCB[PSMSL_RECORD_NUMBER],
728 1668 3 .SCB[PSMSL_PAGE_WIDTH]];
729 1669 3
730 1670 3 ! adjust pointer
731 1671 3 FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
732 1672 3 %CHAR(32), .FUNC_DESC[SIZE]);
733 1673 3
734 1674 3 END;
735 1675 3
736 1676 3 [PSMSK_OPEN]:
737 1677 3 BEGIN
738 1678 3
739 1679 3 GET_FORM_SIZE (.SCB); ! Returns the WidthxLength
740 1680 3
741 1681 3
742 1682 3 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
743 1683 3 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
744 1684 3
745 1685 3 RETURN_IF_ERROR_(ALLOCATE_PAGE( .SCB)); ! Get the page of memory
746 1686 3
747 1687 3 PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
748 1688 3
749 1689 3 ! Always start at top of page
750 1690 3
751 1691 3 PAGE_REF[0,0,.FORM_WIDTH] = PSMSK_CHAR_FF; ! form feed in 0 pos.
752 1692 3
753 1693 3 ! Standard Flag Page 132x66: text covers rows 1 through 58,
754 1694 3 ! translated to frames... ref starts at 1 and length is 57.
755 1695 3 FILL_JOB_FLAG( .SCB,
756 1696 3 PAGE_REF[0,1,.FORM_WIDTH],
757 1697 3 .FORM_WIDTH,
758 1698 3 .FORM_LENGTH - 6 - 2 - 1); ! 6 burst,
759 1699 3 ! 2 spaces before burst,
760 1700 3 ! top margin is 1
761 1701 4 IF (.SEPARATE_FLAG_(JOB_BURST))
762 1702 3 THEN
763 1703 4 BEGIN
764 1704 4 STRING_DESC[SIZE] = %ALLOCATION(BUFFER);
765 1705 4 STRING_DESC[ADDR] = BUFFER;
766 1706 4
767 1707 4 GET_VMS_LOGO
768 1708 4 (.SCB,
769 1709 4 STRING_DESC[0], ! Buffer descriptor
770 1710 4 STRING_DESC[SIZE]); ! Returned length
771 1711 4
772 1712 4 INSERT_FRAME
```

```
773 1713 4 (.SCB,  
774 1714 4 STRING_DESC[0],  
775 1715 4 PAGE_REF[10,.FORM_LENGTH-5,.FORM_WIDTH],  
776 1716 4 .FORM_WIDTH-20, 1);  
777 1717 4 INSERT_FRAME  
778 1718 4 (.SCB,  
779 1719 4 STRING_DESC[0],  
780 1720 4 PAGE_REF[14,.FORM_LENGTH-4,.FORM_WIDTH],  
781 1721 4 .FORM_WIDTH-16, 1);  
782 1722 4 INSERT_FRAME  
783 1723 4 (.SCB,  
784 1724 4 STRING_DESC[0],  
785 1725 4 PAGE_REF[10,.FORM_LENGTH-3,.FORM_WIDTH],  
786 1726 4 .FORM_WIDTH-20, 1);  
787 1727 4 INSERT_FRAME  
788 1728 4 (.SCB,  
789 1729 4 STRING_DESC[0],  
790 1730 4 PAGE_REF[14,.FORM_LENGTH-2,.FORM_WIDTH],  
791 1731 4 .FORM_WIDTH-16, 1);  
792 1732 4 INSERT_FRAME  
793 1733 4 (.SCB,  
794 1734 4 STRING_DESC[0],  
795 1735 4 PAGE_REF[10,.FORM_LENGTH-1,.FORM_WIDTH],  
796 1736 4 .FORM_WIDTH-20, 1);  
797 1737 4 INSERT_FRAME  
798 1738 4 (.SCB,  
799 1739 4 STRING_DESC[0],  
800 1740 4 PAGE_REF[14,.FORM_LENGTH,.FORM_WIDTH],  
801 1741 4 .FORM_WIDTH-16, 1);  
802 1742 3 END;  
803 1743 2 END;  
804 1744 2  
805 1745 2 [PSMSK_CLOSE]: ! Return the Page of Memory  
806 1746 2 RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));  
807 1747 2  
808 1748 2 [OTHERWISE]:  
809 1749 2 RETURN PSM$_FUNNOTSUP;  
810 1750 2  
811 1751 2 TES; ! case .function  
812 1752 2  
813 1753 2 SSS_NORMAL  
814 1754 2  
815 1755 1 END;
```

```
56 0000V CF 007C 00000  
5E FDF8 CE 9E 00002  
54 04 BC D0 0000C  
50 0C BC D0 00010  
05 50 D1 00014  
4F 12 00017  
53 01FC C4 D0 00019  
50 026C C4 D0 0001E
```

```
.ENTRY PSM$JOB_FLAG, Save R2,R3,R4,R5,R6  
MOVAB INSERT_FRAME, R6  
MOVAB -520(SP), SP  
MOVL @SMB_CONTEXT, SCB  
MOVL @FUNCTION, R0  
CMPL R0, #5  
BNEQ 4$  
MOVL 508(SCB), PAGE_REF  
MOVL 620(SCB), R0
```

```
: 1627  
: 1649  
: 1653  
: 1655  
: 1657  
: 1659
```


		01F8	C4	50	D1	00023	CMPL	R0, 504(SCB)		
				11	14	00028	BGTR	1\$		
13		0154	C4	05	E0	0002A	BBS	#5, 340(SCB), 2\$	1661	
51		01F8	C4	06	C3	00030	SUGL3	#6, 504(SCB), R1	1662	
			51	50	D1	00036	CMPL	R0, R1		
				08	19	00039	BLSS	2\$		
		50	00000000G	8F	D0	0003B	1\$:	MOVL	#PSM\$_EOF, R0	1664
					04	00042	RET			
		52		10	AC	D0	2\$:	MOVL	FUNC_DESC, R2	1666
		62	0200	C4	D0	00047	MOVL	512(SCB), (R2)		
		50	0200	C4	C4	0004C	MULL2	512(SCB), R0	1668	
04	A2	50		53	CD	00051	ADDL3	PAGE_REF, R0, 4(R2)		
				62	DD	00056	PUSHL	(R2)	1672	
				20	DD	00058	PUSHL	#32	1671	
			04	A2	DD	0005A	PUSHL	4(R2)		
		0000V	CF	03	FB	0005D	CALLS	#3, DELIMIT_STRING_NOT		
		62		50	D0	00062	MOVL	R0, (R2)		
				00FD	31	00065	3\$:	BRW	9\$	1653
		04		50	D1	00068	4\$:	CMPL	R0, #4	1676
				03	13	0006B	BEQL	5\$		
				00DD	31	0006D	BRW	7\$		
				54	DD	00070	5\$:	PUSHL	SCB	1679
		0000V	CF	01	FB	00072	CALLS	#1, GET_FORM_SIZE		
		55	0200	C4	D0	00077	MOVL	512(SCB), FORM_WIDTH	1682	
		52	01F8	C4	D0	0007C	MOVL	504(SCB), FORM_LENGTH	1683	
				54	DD	00081	PUSHL	SCB	1685	
		0000V	CF	01	FB	00083	CALLS	#1, ALLOCATE_PAGE		
		01		50	E8	00088	BLBS	STATUS, 6\$		
					04	0008B	RET			
		53	01FC	C4	D0	0008C	6\$:	MOVL	508(SCB), PAGE_REF	1687
		63		0C	90	00091	MOVB	#12, (PAGE_REF)	1691	
			F7	A2	9F	00094	PUSHAB	-9(FORM_LENGTH)	1698	
				55	DD	00097	PUSHL	FORM_WIDTH	1697	
				6543	9F	00099	PUSHAB	(FORM_WIDTH)[PAGE_REF]	1696	
				54	DD	0009C	PUSHL	SCB		
		0000V	CF	04	FB	0009E	CALLS	#4, FILL_JOB_FLAG		
BC		0154	C4	05	E1	000A3	BBC	#5, 340(SCB), 3\$	1701	
		6E	0200	8F	3C	000A9	MOVZWL	#512, STRING_DESC	1704	
		04	AE	08	AE	000AE	MOVAB	BUFFER, STRING_DESC+4	1705	
				5E	DD	000B3	PUSHL	SP	1710	
				04	AE	000B5	PUSHAB	STRING_DESC	1709	
				54	DD	000B8	PUSHL	SCB	1708	
		0000V	CF	03	FB	000BA	CALLS	#3, GET_VMS_LOGO		
				01	DD	000BF	PUSHL	#1	1715	
			EC	A5	9F	000C1	PUSHAB	-20(FORM_WIDTH)	1716	
		50	FB	A2	9E	000C4	MOVAB	-5(R2), R0	1715	
		50		55	C4	000C8	MULL2	FORM_WIDTH, R0		
			0A	A043	9F	000CB	PUSHAB	10(R0)[PAGE_REF]		
			0C	AE	9F	000CF	PUSHAB	STRING_DESC	1714	
				54	DD	000D2	PUSHL	SCB	1715	
		66		05	FB	000D4	CALLS	#5, INSERT_FRAME		
				01	DD	000D7	PUSHL	#1	1720	
			F0	A5	9F	000D9	PUSHAB	-16(FORM_WIDTH)	1721	
		50	FC	A2	9E	000DC	MOVAB	-4(R2), R0	1720	
		50		55	C4	000E0	MULL2	FORM_WIDTH, R0		
			0E	A043	9F	000E3	PUSHAB	14(R0)[PAGE_REF]		
			0C	AE	9F	000E7	PUSHAB	STRING_DESC	1719	

66		54	DD	000EA	PUSHL	SCB	1720
		05	FB	000EC	CALLS	#5, INSERT_FRAME	
		01	DD	000EF	PUSHL	#1	1725
	EC	A5	9F	000F1	PUSHAB	-20(FORM_WIDTH)	1726
50	FD	A2	9E	000F4	MOVAB	-3(R2), R0	1725
50		55	C4	000F8	MULL2	FORM_WIDTH, R0	
	OA	A043	9F	000FB	PUSHAB	10(R0)[PAGE_REF]	
	OC	AE	9F	000FF	PUSHAB	STRING_DESC	1724
		54	DD	00102	PUSHL	SCB	1725
66		05	FB	00104	CALLS	#5, INSERT_FRAME	
		01	DD	00107	PUSHL	#1	1730
	FO	A5	9F	00109	PUSHAB	-16(FORM_WIDTH)	1731
50	FE	A2	9E	0010C	MOVAB	-2(R2), R0	1730
50		55	C4	00110	MULL2	FORM_WIDTH, R0	
	OE	A043	9F	00113	PUSHAB	14(R0)[PAGE_REF]	
	OC	AE	9F	00117	PUSHAB	STRING_DESC	1729
		54	DD	0011A	PUSHL	SCB	1730
66		05	FB	0011C	CALLS	#5, INSERT_FRAME	
		01	DD	0011F	PUSHL	#1	1735
	EC	A5	9F	00121	PUSHAB	-20(FORM_WIDTH)	1736
50	FF	A2	9E	00124	MOVAB	-1(R2), R0	1735
50		55	C4	00128	MULL2	FORM_WIDTH, R0	
	OA	A043	9F	0012B	PUSHAB	10(R0)[PAGE_REF]	
	OC	AE	9F	0012F	PUSHAB	STRING_DESC	1734
		54	DD	00132	PUSHL	SCB	1735
66		05	FB	00134	CALLS	#5, INSERT_FRAME	
		01	DD	00137	PUSHL	#1	1740
	FO	A5	9F	00139	PUSHAB	-16(FORM_WIDTH)	1741
52		55	C4	0013C	MULL2	FORM_WIDTH, R2	1740
	OE	A243	9F	0013F	PUSHAB	14(R2)[PAGE_REF]	
	OC	AE	9F	00143	PUSHAB	STRING_DESC	1739
		54	DD	00146	PUSHL	SCB	1740
66		05	FB	00148	CALLS	#5, INSERT_FRAME	
		18	11	0014B	BRB	9\$	1653
02		50	D1	0014D	CMPL	R0, #2	1745
		0B	12	00150	BNEQ	8\$	
		54	DD	00152	PUSHL	SCB	1746
0000V	CF	01	FB	00154	CALLS	#1, DEALLOCATE_PAGE	
	09	50	E8	00159	BLBS	STATUS, 9\$	
		04	00	0015C	RET		
50	00000000G	8F	D0	0015D	MOVL	#PSMS_FUNNOTSUP, R0	1749
		04	00	00164	RET		
50		01	D0	00165	MOVL	#1, R0	1755
		04	00	00168	RET		

; Routine Size: 361 bytes, Routine Base: CODE + 035F

```
817 1756 1 %sbttl 'PSMSJOB_TRAILER - Print a Job Flag Page'
818 1757 1 Functional Description:
819 1758 1 This routine controls the creation of the job trailer page. The
820 1759 1 FUNCTION code dictates the action taken in creation.
821 1760 1 FUNCTION:
822 1761 1 OPEN - Allocate and create the Job Trailer Page
823 1762 1 READ - Return the current line of the Job Trailer Page
824 1763 1 CLOSE - Return the buffer allocated on OPEN
825 1764 1
826 1765 1 Formal Parameters:
827 1766 1 SMB_CONTEXT - Pointer to the SMB
828 1767 1 USER_CONTEXT - User defined pointer (not used here)
829 1768 1 FUNCTION - OPEN, READ, CLOSE
830 1769 1 FUNC_DESC - Pointer to functionally dependent descriptor
831 1770 1 FUNC_ARG - Pointer to functionally dependent argument
832 1771 1
833 1772 1 Implicit Inputs:
834 1773 1 none
835 1774 1
836 1775 1 Implicit Outputs:
837 1776 1 none
838 1777 1
839 1778 1 Returned Value:
840 1779 1 none
841 1780 1
842 1781 1 Side Effects:
843 1782 1 none
844 1783 1 --
845 1784 1 GLOBAL ROUTINE PSMSJOB_TRAILER ( %SBTTL 'JOB_TRAILER'
846 1785 1 SMB_CONTEXT : REF VECTOR,
847 1786 1 USER_CONTEXT : REF VECTOR,
848 1787 1 FUNCTION : REF VECTOR,
849 1788 1 FUNC_DESC : REF VECTOR,
850 1789 1 FUNC_ARG : REF VECTOR
851 1790 1 ) =
852 1791 2 BEGIN
853 1792 2
854 1793 2 LITERAL
855 1794 2 TRAILING = 1;
856 1795 2 LOCAL
857 1796 2 SCB : REF $BLOCK,
858 1797 2 STATUS,
859 1798 2 FORM_WIDTH,
860 1799 2 FORM_LENGTH,
861 1800 2 FORM_SIZE,
862 1801 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
863 1802 2 ! to page
864 1803 2
865 1804 2 SCB = .SMB_CONTEXT[0];
866 1805 2
867 1806 2 ! Check the FUNCTION requested
868 1807 2
869 1808 2 SELECTONEU .FUNCTION[0] OF
870 1809 2 SET
871 1810 2 [PSMSK_READ]:
872 1811 2 BEGIN
873 1812 2 LOCAL TEMP_PTR;
```



```
874 1813
875 1814 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
876 1815
877 1816 IF .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 2 - 6)
878 1817 THEN
879 1818     RETURN PSMS_EOF;
880 1819
881 1820 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
882 1821 FUNC_DESC[ADDR] = PAGE_REF[0, .SCB[PSMSL_RECORD_NUMBER],
883 1822     .SCB[PSMSL_PAGE_WIDTH]];
884 1823
885 1824     ! adjust pointer
886 1825 FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
887 1826     %CHAR(32), .FUNC_DESC[SIZE]);
888 1827
889 1828 END;
890 1829
891 1830 [PSMSK_OPEN]:
892 1831 BEGIN
893 1832
894 1833 GET_FORM_SIZE (.SCB);           ! Returns the WidthxLength
895 1834
896 1835
897 1836 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
898 1837 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
899 1838
900 1839 RETURN_IF_ERROR(ALLOCATE_PAGE(.SCB));   ! Get the page of memory
901 1840
902 1841 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];    ! My local page pointer
903 1842
904 1843 ! Always start at top of page
905 1844
906 1845 PAGE_REF[0,0,.FORM_WIDTH] = PSMSK_CHAR_FF;   ! form feed in 0 pos.
907 1846
908 1847 ! Standard Trailer Page 132x66: text covers rows 1 through 58,
909 1848 ! translated to frames... ref starts at 1 and length is 57.
910 1849 FILL_JOB_TRAILER(.SCB,
911 1850     PAGE_REF[0,1,.FORM_WIDTH],
912 1851     .FORM_WIDTH,
913 1852     .FORM_LENGTH - 6 - 2 - 1);   ! ...6 burst, 2 sp,
914 1853     ! top margin is 1
915 1854
916 1855 END;
917 1856 [PSMSK_CLOSE]:           ! Return the Page of Memory
918 1857     RETURN_IF_ERROR(DEALLOCATE_PAGE(.SCB));
919 1858
920 1859 [OTHERWISE]:
921 1860     RETURN PSMS_FUNNOTSUP;
922 1861
923 1862 TES; ! case .function
924 1863
925 1864 SS$_NORMAL
926 1865 1 END;
```

				003C	00000	.ENTRY	PSM\$JOB TRAILER, Save R2,R3,R4,R5	1784
		52	04	BC	D0	MOVL	@SMB_CONTEXT, SCB	1804
		50	0C	BC	D0	MOVL	@FUNCTION, R0	1808
		05		50	D1	CMPL	R0, #5	1810
				41	12	BNEQ	2\$	
		54	01FC	C2	D0	MOVL	508(SCB), PAGE_REF	1814
50	01F8	C2		08	C3	SUBL3	#8, 504(SCB), R0	1816
		50	026C	C2	D1	CMPL	620(SCB), R0	
				08	15	BLEQ	1\$	
		50	00000000G	8F	D0	MOVL	#PSM\$_EOF, R0	1818
					04	RET		
		53	10	AC	D0	MOVL	FUNC_DESC, R3	1820
		63	0200	C2	D0	MOVL	512(SCB), (R3)	
04	50	026C	0200	C2	C5	MULL3	512(SCB), 620(SCB), R0	1822
A3		50		54	C1	ADDL3	PAGE_REF, R0, 4(R3)	
				63	DD	PUSHL	(R3)	1826
				20	DD	PUSHL	#32	1825
			04	A3	DD	PUSHL	4(R3)	
	0000V	CF		03	FB	CALLS	#3, DELIMIT_STRING_NOT	
		63		50	D0	MOVL	R0, (R3)	
				51	11	BRB	5\$	1808
		04		50	D1	CMPL	R0, #4	1830
				34	12	BNEQ	3\$	
				52	DD	PUSHL	SCB	1833
	0000V	CF		01	FB	CALLS	#1, GET_FORM_SIZE	
		55	0200	C2	D0	MOVL	512(SCB), FORM_WIDTH	1836
		53	01F8	C2	D0	MOVL	504(SCB), FORM_LENGTH	1837
				52	DD	PUSHL	SCB	1839
	0000V	CF		01	FB	CALLS	#1, ALLOCATE_PAGE	
		34		50	E9	BLBC	STATUS, 6\$	
		54	01FC	C2	D0	MOVL	508(SCB), PAGE_REF	1841
		64		0C	90	MOVB	#12, (PAGE_REF)	1845
			F7	A3	9F	PUSHAB	-9(FORM_LENGTH)	1852
				55	DD	PUSHL	FORM_WIDTH	1851
				6544	9F	PUSHAB	(FORM_WIDTH)[PAGE_REF]	1850
				52	DD	PUSHL	SCB	
	0000V	CF		04	FB	CALLS	#4, FILL_JOB_TRAILER	
				18	11	BRB	5\$	1808
		02		50	D1	CMPL	R0, #2	1855
				0B	12	BNEQ	4\$	
				52	DD	PUSHL	SCB	1856
	0000V	CF		01	FB	CALLS	#1, DEALLOCATE_PAGE	
		09		50	E8	BLBS	STATUS, 5\$	
				04		RET		
		50	00000000G	8F	D0	MOVL	#PSM\$_FUNNOTSUP, R0	1859
					04	RET		
		50		01	D0	MOVL	#1, R0	1865
				04	000A4	RET		

; Routine Size: 165 bytes, Routine Base: CODE + 04C8

```
928 1866 1 %sbtll 'PSMSPAGE_HEADER - Print a Header at the Top of each Page'
929 1867 1 Functional Description:
930 1868 1 Creates a page header for the current file and prints it at the
931 1869 1 top of each page.
932 1870 1 FUNCTION:
933 1871 1 OPEN - Allocate and create the Page Header
934 1872 1 READ - Return the current header with the new page number
935 1873 1 CLOSE - Deallocate the header
936 1874 1
937 1875 1 Formal Parameters:
938 1876 1 SMB_CONTEXT - Pointer to the SMB
939 1877 1 USER_CONTEXT - User defined pointer (not used here)
940 1878 1 FUNCTION - OPEN, READ, CLOSE
941 1879 1 FUNC_DESC - Pointer to functionally dependent descriptor
942 1880 1 FUNC_ARG - Pointer to functionally dependent argument
943 1881 1
944 1882 1 Implicit Inputs:
945 1883 1 none
946 1884 1
947 1885 1 Implicit Outputs:
948 1886 1 none
949 1887 1
950 1888 1 Returned Value:
951 1889 1 none
952 1890 1
953 1891 1 Side Effects:
954 1892 1 none
955 1893 1 --
956 1894 1 GLOBAL ROUTINE PSMSPAGE_HEADER ( %SBTLL 'PAGE_HEADER'
957 1895 1 SMB_CONTEXT : REF VECTOR,
958 1896 1 USER_CONTEXT : REF VECTOR,
959 1897 1 FUNCTION : REF VECTOR,
960 1898 1 FUNC_DESC : REF VECTOR,
961 1899 1 FUNC_ARG : REF VECTOR
962 1900 1 ) =
963 1901 2 BEGIN
964 1902 2 LOCAL
965 1903 2 SCB : REF $BLOCK;
966 1904 2
967 1905 2 SCB = .SMB_CONTEXT[0];
968 1906 2
969 1907 2 ! Check the FUNCTION requested
970 1908 2
971 1909 2 SELECTONEU .FUNCTION[0] OF
972 1910 2 SET
973 1911 2 [PSMSK_READ]:
974 1912 2 BEGIN
975 1913 2
976 1914 2 IF .SCB[PSMSL_RECORD_NUMBER] GTRU 0
977 1915 2 THEN
978 1916 2 RETURN PSMS_EOF;
979 1917 2
980 1918 2 ! Use the supplied string descriptor as a temp for the page number
981 1919 2
982 1920 2 FUNC_DESC[SIZE] = 5;
983 1921 2 FUNC_DESC[ADDR] = .SCB_ADDR_(PAGE_HEADER) + .SCB_SIZE_(PAGE_HEADER) - 8;
984 1922 2 !
```



```

985      1923      | Write the page number into the end of the page header buffer
986      1924      | (note -- since the page number can decrease we always fill out
987      1925      | the page number area with blanks to overwrite any prior data)
988      1926
989      1927      $FAO (
990      1928          $DESCRIPTOR ('!5<!UL!>' ),      | pad with trailing spaces
991      1929          FUNC_DESC[SIZE],                  | ignore return length
992      1930          FUNC_DESC[0],                      | temp output buffer desc
993      1931          .SCB[PSM$L_PAGE]                  | current page number
994      1932      );
995      1933
996      1934      | copy the page header descriptor size and address to
997      1935      | the function descriptor
998      1936
999      1937      FUNC_DESC[SIZE] = .SCB_SIZE_ (PAGE_HEADER);
1000     1938      FUNC_DESC[ADDR] = .SCB_ADDR_ (PAGE_HEADER);
1001     1939
1002     1940      END;
1003     1941
1004     1942      [PSM$K_OPEN]:
1005     1943      BEGIN
1006     1944          | set carriage control to imbedded
1007     1945
1008     1946          FUNC_ARG[0] = PSM$K_CC_INTERNAL;
1009     1947
1010     1948          | Format everything but the page number, but only do it once per task
1011     1949
1012     1950          IF TESTBITCS (SCB[PSM$V_PAGE_HEADER_BUILT])
1013     1951          THEN
1014     1952              CREATE_PAGE_HEADER (.SCB);
1015     1953
1016     1954          END;
1017     1955
1018     1956      [PSM$K_START_TASK]:
1019     1957          | Set the size of the page header equal to the page width adjusted
1020     1958          | for margins.
1021     1959
1022     1960          BEGIN
1023     1961              GET_FORM_SIZE (.SCB);                  ! Returns the WidthxLength
1024     1962
1025     1963              | Adjust for margins and imbedded carriage control
1026     1964
1027     1965              SCB[PSM$L_PAGE_WIDTH] = .SCB[PSM$L_PAGE_WIDTH]
1028     1966                  - .SCB[PSM$L_LEFT_MARGIN]      | less leading spaces
1029     1967                  - .SCB[PSM$L_RIGHT_MARGIN]     | less early truncation
1030     1968                  + 3;                             | plus trailing carr
1031     1969                  | cntrl <CR><LF><LF>
1032     1970
1033     1971              RETURN_IF_ERROR_ (STR$GET1_DX (XREF (.SCB[PSM$L_PAGE_WIDTH]),
1034     1972                  SCB[PSM$Q_PAGE_HEADER]));
1035     1973
1036     1974          END;
1037     1975
1038     1976      [OTHERWISE]:
1039     1977          RETURN PSM$FUNNOTSUP;
1040     1978
1041     1979      TES; ! case .function
```

SEPARATE
V04-001

Print Symbiont -- separation routines
PAGE_HEADER

H 15
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 30
(10)

: 1042
: 1043
: 1044
1980 2 SS\$_NORMAL
1981 2
1982 1 END;

3E	21	4C	55	21	3C	35	21	0056D	P.AAB:	.ASCII	\!5<!UL!>\	:
								00575		.BLKB	3	:
						00000008		00578	P.AAA:	.LONG	8	:
						00000000		0057C		.ADDRESS	P.AAB	:
										.EXTRN	SYSS\$FAO	:
										.ENTRY	PSM\$PAGE_HEADER, Save R2,R3,R4	: 1894
										SUBL2	#4, SP	: 1905
										MOVL	@SMB_CONTEXT, SCB	: 1909
										MOVL	@FUNCTION, R0	: 1911
										CMPL	R0, #5	: 1914
										BNEQ	2\$: 1916
										TSTL	620(SCB)	: 1920
										BEQL	1\$: 1921
										MOVL	#PSM\$_EOF, R0	: 1932
										RET		:
										MOVL	FUNC_DESC, R3	: 1937
										MOVL	#5, (R3)	: 1938
										MOVL	SCB, R4	: 1909
										MOVZWL	496(R4), R0	: 1942
										ADDL2	500(SCB), R0	:
										MOVAB	-8(R0), 4(R3)	: 1946
										PUSHL	492(SCB)	: 1950
										PUSHL	R3	: 1952
										PUSHL	R3	:
										PUSHAB	P.AAA	: 1909
										CALLS	#4, SYSS\$FAO	: 1956
										MOVZWL	496(R4), (R3)	: 1961
										MOVL	500(SCB), 4(R3)	: 1965
										BRB	5\$: 1966
										CMPL	R0, #4	: 1967
										BNEQ	3\$: 1968
										MOVL	#1, @FUNC_ARG	: 1972
										BBSS	#8, 16(SCB), 5\$:
										PUSHL	SCB	: 1976
										CALLS	#1, CREATE_PAGE_HEADER	: 1977
										BRB	5\$: 1978
										CMPL	R0, #16	: 1979
										BNEQ	4\$: 1980
										PUSHL	SCB	: 1981
										CALLS	#1, GET_FORM_SIZE	: 1982
										MOVAB	512(SCB), R1	: 1983
										SUBL3	188(SCB), (R1), R0	: 1984
										SUBL2	328(SCB), R0	: 1985
										MOVAB	3(R0), (R1)	: 1986
										PUSHAB	496(SCB)	: 1987
										MOVL	(R1), 4(SP)	: 1988
										PUSHAB	4(SP)	: 1989
										CALLS	#2, STR\$GET1_DX	: 1990
										BLBS	STATUS, 5\$: 1991

SEPARATE
V04-001

Print Symbiont -- separation routines
PAGE_HEADER

1 15
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 31
(10)

50 00000000G	8F	04 000A4	RET	
		D0 000A5	4\$: MOVL	#PSMS_FUNNOTSUP, R0
		04 000AC	RET	
50	01	D0 000AD	5\$: MOVL	#1, R0
		04 000B0	RET	

: 1976
: 1982
:

: Routine Size: 177 bytes, Routine Base: CODE + 0580

```
1046 1983 1 ROUTINE PARSE_FILE_NAME ( %SBTTL 'PARSE_FILE_NAME'
1047 1984 1 FILENAME : REF $BBLOCK,
1048 1985 1 ITEM_CODE
1049 1986 1 RESULT : REF VECTOR
1050 1987 1 ) =
1051 1988 2 BEGIN
1052 1989 2
1053 1990 2 LOCAL
1054 1991 2 LIST : $ITMBLK [1,8]
1055 1992 2 ;
1056 1993 2
1057 1994 2 CH$FILL (0, %ALLOCATION (LIST), LIST);
1058 1995 2
1059 1996 2 LIST [0, ITMSW_ITMCD] = .ITEM_CODE;
1060 1997 2
1061 1998 2 RETURN_IF_ERROR_ ($FILESCAN (SRCSTR=.FILENAME, VALUELST=LIST));
1062 1999 2
1063 2000 2 RESULT[SIZE] = .LIST[0, ITMSW_BUFSIZ];
1064 2001 2 RESULT[ADDR] = .LIST[0, ITMSL_BUFADR];
1065 2002 2
1066 2003 2 SS$_NORMAL
1067 2004 2
1068 2005 1 END;
```

.EXTRN SYSS\$FILESCAN

003C 00000 PARSE_FILE_NAME:

0C	00	5E	0C	C2	00002	.WORD	Save R2,R3,R4,R5	1983
		6E	00	2C	000C5	SUBL2	#12, SP	1994
			6E		0000A	MOVCS	#0, (SP), #0, #12, LIST	1996
	02	AE	08	AC	80 0000B	MOVW	ITEM_CODE, LIST+2	1998
			04	7E	D4 00010	CLRL	-(SP)	
			04	AE	9F 00012	PUSHAB	LIST	
			04	AC	DD 00015	PUSHL	FILENAME	
	00000000G	00	03	FB	00018	CALLS	#3, SYSS\$FILESCAN	
		0F	50	E9	0001F	BLBC	STATUS, 1\$	
		50	0C	AC	D0 00022	MOVL	RESULT, R0	2000
		60	6E	3C	00026	MOVZWL	LIST, (R0)	
	04	A0	04	AE	D0 00029	MOVL	LIST+4, 4(R0)	2001
		50	01	D0	0002E	MOVL	#1, R0	2005
			04	00031	1\$:	RET		

; Routine Size: 50 bytes, Routine Base: CODE + 0631


```
1070 2006 1 %sbttl 'ALLOCATE_PAGE - Allocate the Page of Memory'
1071 2007 1 ++
1072 2008 1 Functional Description:
1073 2009 1 This routine allocates memory in an amount of
1074 2010 1 memory equal to the largest Form Size supported.
1075 2011 1
1076 2012 1 Formal Parameters:
1077 2013 1 SCB - Address of the SCB
1078 2014 1
1079 2015 1 Implicit Inputs:
1080 2016 1 none
1081 2017 1
1082 2018 1 Implicit Outputs:
1083 2019 1 none
1084 2020 1
1085 2021 1 Returned Value:
1086 2022 1 none
1087 2023 1
1088 2024 1 Side Effects:
1089 2025 1 none
1090 2026 1 --
1091 2027 1 ROUTINE ALLOCATE_PAGE(
1092 2028 1 SCB : REF $BBLOCK
1093 2029 1 ) =
1094 2030 2 BEGIN
1095 2031 2 LOCAL
1096 2032 2 PAGE_SIZE;
1097 2033 2
1098 2034 2 PAGE_SIZE = .SCB[PSM$L_PAGE_WIDTH] * (.SCB[PSM$L_PAGE_LENGTH]+1);
1099 2035 2
1100 P 2036 2 RETURN_IF_ERROR_( LIB$GET_VM ( %REF(.PAGE_SIZE),
1101 2037 2 SCB[PSM$A_PAGE_POINTER]));
1102 2038 2 ! Fill it with Blanks
1103 2039 2 CH$FILL (%CHAR(32), .PAGE_SIZE,
1104 2040 2 .SCB[PSM$A_PAGE_POINTER]);
1105 2041 2
1106 2042 2 RETURN SS$NORMAL;
1107 2043 1 END;
```

				003C 00000 ALLOCATE_PAGE:		
		5E		04 C2 00002	WORD	Save R2,R3,R4,R5
		53	04	AC D0 00005	SUBL2	#4, SP
	50	01F8	C3	01 C1 00009	MOVL	SCB, R3
	52		50	0200 C3 C5 0000F	ADDL3	#1, 504(R3), R0
			01FC	C3 9F 00015	MULL3	512(R3), R0, PAGE_SIZE
		04	AE	52 D0 00019	PUSHAB	508(R3)
			04	AE 9F 0001D	MOVL	PAGE_SIZE, 4(SP)
		00000000G	00	02 FB 00020	PUSHAB	4(SP)
			0B	50 E9 00027	CALLS	#2, LIB\$GET_VM
52	20	6E		00 2C 0002A	BLBC	STATUS, 1\$
			01FC	D3 0002F	MOVCS	#0, (SP), #32, PAGE_SIZE, a508(R3)
		50		01 D0 00032	MOVL	#1, R0

2027
2034
2037
2040
2042

SEPARATE
V04-001

Print Symbiont -- separation routines
ALLOCATE_PAGE - Allocate the Page of Memory

L 15
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 34
(12)
; 2043

04 00035 1\$: RET
; Routine Size: 54 bytes, Routine Base: CODE + 0663

```
1109 2044 1 %sbtll 'DEALLOCATE_PAGE - Deallocate the Page of Memory'
1110 2045 1 ++
1111 2046 1 Functional Description:
1112 2047 1 This routine deallocates memory in an amount of
1113 2048 1 memory equal to the largest form Size supported.
1114 2049 1
1115 2050 1 Formal Parameters:
1116 2051 1 SCB - Address of the SCB
1117 2052 1
1118 2053 1 Implicit Inputs:
1119 2054 1 none
1120 2055 1
1121 2056 1 Implicit Outputs:
1122 2057 1 none
1123 2058 1
1124 2059 1 Returned Value:
1125 2060 1 none
1126 2061 1
1127 2062 1 Side Effects:
1128 2063 1 none
1129 2064 1 --
1130 2065 1 ROUTINE DEALLOCATE_PAGE (
1131 2066 1 SCB : REF $BBLOCK
1132 2067 1 ) =
1133 2068 2 BEGIN
1134 2069 2 LOCAL
1135 2070 2 PAGE_SIZE;
1136 2071 2
1137 2072 2 PAGE_SIZE = .SCB[PSM$PAGE_WIDTH] * (.SCB[PSM$PAGE_LENGTH]+1);
1138 2073 2
1139 P 2074 2 RETURN_IF_ERROR_( LIB$FREE_VM ( %REF(.PAGE_SIZE),
1140 2075 2 SCB[PSM$PAGE_POINTER]));
1141 2076 2
1142 2077 2 RETURN SS$_NORMAL;
1143 2078 1 END;
```

0000 00000 DEALLOCATE PAGE:

	5E		04	C2	00002	WORD	Save nothing	
	51		04	AC	D0 00005	SUBL2	#4, SP	
50	01F8	C1		01	C1 00009	MOVL	SCB, R1	
	50		0200	C1	C4 0000F	ADDL3	#1, 504(R1), R0	
			01FC	C1	9F 00014	MULL2	512(R1), PAGE_SIZE	
	04	AE		50	D0 00018	PUSHAB	508(R1)	
			04	AE	9F 0001C	MOVL	PAGE_SIZE, 4(SP)	
00000000G	00			02	FB 0001F	PUSHAB	4(SP)	
	03			50	E9 00026	CALLS	#2, LIB\$FREE_VM	
	50			01	D0 00029	BLBC	STATUS, 1\$	
				04	0002C 1\$:	MOVL	#1, R0	
						RET		

; Routine Size: 45 bytes, Routine Base: CODE + 0699

: 2065
: 2072
: 2075
: 2077
: 2078

```
1145 2079 1 %sbttl 'CREATE_PAGE_HEADER - Allocate and Format the Page Header'
1146 2080 1 ++
1147 2081 1 Functional Description:
1148 2082 1 This routine allocates memory and formats the information
1149 2083 1 for the page header. Returns success if allocation of memory
1150 2084 1 was successful.
1151 2085 1
1152 2086 1 Formal Parameters:
1153 2087 1 SCB - Address of the SCB
1154 2088 1
1155 2089 1 Implicit Inputs:
1156 2090 1 none
1157 2091 1
1158 2092 1 Implicit Outputs:
1159 2093 1 none
1160 2094 1
1161 2095 1 Returned Value:
1162 2096 1 none
1163 2097 1
1164 2098 1 Side Effects:
1165 2099 1 none
1166 2100 1 --
1167 2101 1 ROUTINE CREATE_PAGE_HEADER (
1168 2102 1 SCB : REF $BBLOCK
1169 2103 1 ) =
1170 2104 1 BEGIN
1171 2105 1
1172 2106 1 LOCAL
1173 2107 1 REMAINING, ! Remaining header space
1174 2108 1 NAME_LENGTH, ! Trimmed file name length
1175 2109 1 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes0
1176 2110 1 STR_DESC : VECTOR [2],
1177 2111 1 HEADER_REF : REF PAGE_ARRAY,
1178 2112 1 HEADER_SIZE :
1179 2113 1
1180 2114 1 !! SMALL WIDTHS -- THE PAGE NUMBER SHOULD BE THE ONLY THING PRINTED
1181 2115 1 !! WHEN THE WIDTH IS TOO SMALL. DATE vs. FILENAME IS DEVO'S CHOICE
1182 2116 1
1183 2117 1 HEADER_SIZE = .SCB_SIZE_ (PAGE_HEADER) - 3; ! don't include the carriage
1184 2118 1 ! control area of 3 bytes
1185 2119 1 HEADER_REF = .SCB_ADDR_ (PAGE_HEADER);
1186 2120 1
1187 2121 1 CH$FILL (%CHAR(32), .HEADER_SIZE, .HEADER_REF);
1188 2122 1
1189 2123 1 ! Insert imbedded carriage control <LF><LF><CR>
1190 2124 1
1191 2125 1 CH$FILL (PSMSK_CHAR_LF, 2, (.HEADER_REF + .HEADER_SIZE));
1192 2126 1 CH$FILL (PSMSK_CHAR_CR, 1, (.HEADER_REF + .HEADER_SIZE) + 2);
1193 2127 1 ! address is offset by two
1194 2128 1
1195 2129 1 ! If the header is too small even for 'Page 99999' then disable page
1196 2130 1 ! headers. (Maybe this code should be in 'MESSAGE').
1197 2131 1
1198 2132 1 IF .HEADER_SIZE LSSU 10
1199 2133 1 THEN
1200 2134 1 RETURN SS$_NORMAL;
1201 2135 1
```



```
1202 2136 2 ! Set up the buffer descriptor for "GET_xxx" Routines
1203 2137 2
1204 2138 2 STR_DESC[SIZE] = %ALLOCATION(BUFFER); ! allocate for routines
1205 2139 2 STR_DESC[ADDR] = BUFFER; ! init address
1206 2140 2
1207 2141 2 ! Insert the word "Page "
1208 2142 2
1209 2143 2 MOVE_FRAME
1210 2144 2 (.SCB,
1211 2145 2 $DESCRIPTOR ('Page '),
1212 2146 2 HEADER_REF[.HEADER_SIZE-10,0,.SCB[PSMSL_FORM_WIDTH]],
1213 2147 2 5,
1214 2148 2 1);
1215 2149 2
1216 2150 2 ! Get the filename - include the expected length
1217 2151 2
1218 2152 2 STR_DESC[SIZE] = %ALLOCATION(BUFFER); ! reset buffer length
1219 2153 2 GET_FILE_NAME
1220 2154 2 (.SCB, ! SCB addr.
1221 2155 2 .HEADER_SIZE - 11, ! expected length (less page #)
1222 2156 2 STR_DESC[0], ! Buffer descriptor
1223 2157 2 STR_DESC[SIZE]); ! Returned length
1224 2158 2
1225 2159 2 NAME_LENGTH = .STR_DESC[SIZE] + 1; ! Save the trimmed length
1226 2160 2
1227 2161 2 INSERT_FRAME ! Left Justified
1228 2162 2 (.SCB,
1229 2163 2 STR_DESC[0],
1230 2164 2 HEADER_REF[0,0,.SCB[PSMSL_FORM_WIDTH]],
1231 2165 2 .NAME_LENGTH - 1, ! Always less than frame_width
1232 2166 2 1);
1233 2167 2
1234 2168 2
1235 2169 2 ! The area remaining for the date is the original header
1236 2170 2 ! width less the size of the file name, less the size for the page
1237 2171 2 ! number field ('Page 99999') less one blank for each.
1238 2172 2
1239 2173 2 REMAINING = .HEADER_SIZE - .NAME_LENGTH - 10 - 1;
1240 2174 2 IF .REMAINING LESS 18
1241 2175 2 THEN
1242 2176 2 RETURN SSS_NORMAL;
1243 2177 2
1244 2178 2 ! Get the file revision date and center it between file name and page number
1245 2179 2
1246 2180 2 STR_DESC[SIZE] = %ALLOCATION(BUFFER); ! reset buffer length
1247 2181 2 GET_REVISION_DATE
1248 2182 2 (.SCB, ! SCB addr.
1249 2183 2 STR_DESC[0], ! Buffer descriptor
1250 2184 2 STR_DESC[SIZE]); ! Returned length
1251 2185 2
1252 2186 2 CENTER_FRAME
1253 2187 2 (.SCB,
1254 2188 2 STR_DESC[0],
1255 2189 2 HEADER_REF[.NAME_LENGTH,0,.SCB[PSMSL_FORM_WIDTH]],
1256 2190 2 .REMAINING,
1257 2191 2 1);
1258 2192 2
```

: 1259
: 1260
: 1261
2193 2 RETURN SSS_NORMAL;
2194 2
2195 1 END;

20 65 67 61 50 006C6 P.AAD: .ASCII \Page \
006CB .BLKB 1
00000005 006CC P.AAC: .LONG 5
00000000 006D0 .ADDRESS P.AAD

01FC 00000 CREATE_PAGE_HEADER:

5E	FDF8	CE	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8	2101
58	04	AC	D0	00007	MOVAB	-520(SP), SP	2117
50	01F0	C8	9E	0000B	MOVAB	496(R8), R0	
56		60	3C	00010	MOVZWL	(R0), HEADER_SIZE	
56		03	C2	00013	SUBL2	#3, HEADER_SIZE	
57	04	A0	D0	00016	MOVL	4(R0), HEADER_REF	2119
6E		00	2C	0001A	MOVCS	#0, (SP), #32, HEADER_SIZE, (HEADER_REF)	2121
		67		0001F			
		6647	9F	00020	PUSHAB	(HEADER_SIZE)[HEADER_REF]	2125
9E	0A0A	8F	B0	00023	MOVW	#2570, 5(SP)+	
57		56	C1	00028	ADDL3	HEADER_SIZE, HEADER_REF, R0	
02	A0	0D	90	0002C	MOVB	#13, 2(R0)	2126
0A		56	D1	00030	CMPL	HEADER_SIZE, #10	2132
		72	1F	00033	BLSSU	1\$	
6E	0200	8F	3C	00035	MOVZWL	#512, STR_DESC	2138
04	AE	08	AE	0003A	MOVAB	BUFFER, STR_DESC+4	2139
		01	DD	0003F	PUSHL	#1	2146
		05	DD	00041	PUSHL	#5	
	F6	A647	9F	00043	PUSHAB	-10(HEADER_SIZE)[HEADER_REF]	
	AE	AF	9F	00047	PUSHAB	P.AAC	2145
		58	DD	0004A	PUSHL	R8	2146
0000V	CF	05	FB	0004C	CALLS	#5, MOVE_FRAME	
6E	0200	8F	3C	00051	MOVZWL	#512, STR_DESC	2152
		5E	DD	00056	PUSHL	SP	2157
	04	AE	9F	00058	PUSHAB	STR_DESC	2156
	F5	A6	9F	0005B	PUSHAB	-11(HEADER_SIZE)	2155
		58	DD	0005E	PUSHL	R8	2154
0000V	CF	04	FB	00060	CALLS	#4, GET_FILE_NAME	
52	6E	01	C1	00065	ADDL3	#1, STR_DESC, NAME_LENGTH	2159
		01	DD	00069	PUSHL	#1	2164
	FF	A2	9F	0006B	PUSHAB	-1(NAME_LENGTH)	2165
		57	DD	0006E	PUSHL	HEADER_REF	2164
	0C	AE	9F	00070	PUSHAB	STR_DESC	2163
		58	DD	00073	PUSHL	R8	2164
0000V	CF	05	FB	00075	CALLS	#5, INSERT_FRAME	
56		52	C2	0007A	SUBL2	NAME_LENGTH, R6	2173
56		0B	C2	0007D	SUBL2	#11, REMAINING	
12		56	D1	00080	CMPL	REMAINING, #18	2174
		22	1F	00083	BLSSU	1\$	
6E	0200	8F	3C	00085	MOVZWL	#512, STR_DESC	2180
		5E	DD	0008A	PUSHL	SP	2184
	04	AE	9F	0008C	PUSHAB	STR_DESC	2183

SEPARATE
V04-001

Print Symbiont -- separation routines
CREATE_PAGE_HEADER - Allocate and format the Pa

D 16
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 39
(14)

0000V	CF	58	DD	0008F	PUSHL	R8	:	2182
		03	FB	00091	CALLS	#3, GET_REVISION_DATE	:	
		01	DD	00096	PUSHL	#1	:	2189
		56	DD	00098	PUSHL	REMAINING	:	2190
		6247	9F	0009A	PUSHAB	(NAME_LENGTH)[HEADER_REF]	:	2189
	CC	AE	9F	0009D	PUSHAB	STR_DESC	:	2188
		58	DD	000A0	PUSHL	R8	:	2189
0000V	CF	05	FB	000A2	CALLS	#5, CENTER_FRAME	:	
	50	01	DD	000A7	MOVL	#1, R0	:	2193
		04	DD	000AA	RET		:	2195

: Routine Size: 171 bytes, Routine Base: CODE + 06D4

```
1263 2196 1 %sbttl 'FILL_FILE_FLAG - Insert Information into the FILE Page'
1264 2197 1 ++
1265 2198 1 Functional Description:
1266 2199 1 This procedure controls all inserts required for the FILE Page.
1267 2200 1
1268 2201 1 Formal Parameters:
1269 2202 1     SCB - Address of the SCB
1270 2203 1     PAGE_REF - Pointer to the Page (first byte)
1271 2204 1     PAGE_LENGTH - Length of Frame
1272 2205 1     PAGE_WIDTH - Width of Frame
1273 2206 1
1274 2207 1 Implicit Inputs:
1275 2208 1     none
1276 2209 1
1277 2210 1 Implicit Outputs:
1278 2211 1     none
1279 2212 1
1280 2213 1 Returned Value:
1281 2214 1     none
1282 2215 1
1283 2216 1 Side Effects:
1284 2217 1     none
1285 2218 1 --
1286 2219 1 ROUTINE FILL_FILE_FLAG (
1287 2220 1     SCB : REF $BBLOCK,
1288 2221 1     PAGE_REF : REF PAGE_ARRAY,
1289 2222 1     PAGE_WIDTH,
1290 2223 1     PAGE_LENGTH
1291 2224 1 ): NOVALUE =
1292 2225 1 BEGIN
1293 2226 2
1294 2227 2 LITERAL K_MAX_BUFFER_SIZE = 512;
1295 2228 2
1296 2229 2 LOCAL
1297 2230 2     RET_LEN : VECTOR[1],
1298 2231 2     TOP_OFFSET,
1299 2232 2     BOTTOM_OFFSET,
1300 2233 2     BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
1301 2234 2     STRING_DESC : VECTOR [2]; ! Descriptor to current string
1302 2235 2
1303 2236 2 ! Allocate the buffer for "GET_xxx" Routines
1304 2237 2
1305 2238 2 STRING_DESC[SIZE] = %ALLOCATION(BUFFER); ! allocate for routines
1306 2239 2 STRING_DESC[ADDR] = BUFFER; ! init address
1307 2240 2
1308 2241 2 TOP_OFFSET = 0; ! start insert at zero
1309 2242 2 BOTTOM_OFFSET = .PAGE_LENGTH - 2; ! Note: offset includes next
1310 2243 2 ! "insert" frame length
1311 2244 2
1312 2245 2 ! Burst characters
1313 2246 2
1314 2247 2 FILL_FRAME (.SCB
1315 2248 2     .SCB[PSMSB_FILE_BURST_CHAR],
1316 2249 2     PAGE_REF[0],TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
1317 2250 2
1318 2251 2 FILL_FRAME (.SCB
1319 2252 2     .SCB[PSMSB_FILE_BURST_CHAR],
```



```
1320 2253 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
1321 2254 2
1322 2255 2 FILL_FRAME (.SCB,
1323 2256 2 SCB,
1324 2257 2 PAGE_REF[10,.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);
1325 2258 2
1326 2259 2 FILL_FRAME (.SCB,
1327 2260 2 SCB,
1328 2261 2 PAGE_REF[10,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);
1329 2262 2
1330 2263 2 FILL_FRAME (.SCB,
1331 2264 2 SCB[PSMSB_JOB BURST CHAR],
1332 2265 2 PAGE_REF[14,.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-28, 3);
1333 2266 2
1334 2267 2 FILL_FRAME (.SCB,
1335 2268 2 SCB[PSMSB_JOB BURST CHAR],
1336 2269 2 PAGE_REF[14,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-28, 3);
1337 2270 2
1338 2271 2 Get the sys$announce note and output to page
1339 2272 2 note: system announcement will fit or will be truncated so there is
1340 2273 2 no updating of "offsets"
1341 2274 2
1342 2275 2 ! re-init
1343 2276 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1344 2277 2
1345 2278 2 GET_SYSTEM_ANNOUNCEMENT
1346 2279 2 (.SCB, ! SCB addr.
1347 2280 2 STRING_DESC[0], ! Buffer descriptor
1348 2281 2 STRING_DESC[SIZE]); ! Returned length
1349 2282 2
1350 2283 2 CENTER_FRAME (.SCB,
1351 2284 2 STRING_DESC[0],
1352 2285 2 PAGE_REF[0,.TOP_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);
1353 2286 2
1354 2287 2 ! re-init
1355 2288 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1356 2289 2
1357 2290 2 GET_DIGITAL_LOGO
1358 2291 2 (.SCB, ! SCB addr.
1359 2292 2 STRING_DESC[0], ! Buffer descriptor
1360 2293 2 STRING_DESC[SIZE]); ! Returned length
1361 2294 2
1362 2295 2 CENTER_FRAME (.SCB,
1363 2296 2 STRING_DESC[0],
1364 2297 2 PAGE_REF[0,.BOTTOM_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);
1365 2298 2
1366 2299 2
1367 2300 2 Create a sentence describing the current job.
1368 2301 2
1369 2302 2 ! re-init
1370 2303 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE, ! reset buffer size
1371 2304 2 TOP_OFFSET = .TOP_OFFSET + 4; ! adjust & allow for spacing
1372 2305 2
1373 2306 2 GET_JOB_DESCRIPTION
1374 2307 2 (.SCB, ! SCB addr.
1375 2308 2 ! Use present tense
1376 2309 2 STRING_DESC[0], ! Buffer descriptor
```

```
1377 2310 2 STRING_DESC[SIZE]); ! Returned length
1378 2311 2
1379 2312 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1380 2313 2 (.SCB,
1381 2314 2 STRING_DESC[0], ! string ref.
1382 2315 2 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
1383 2316 2 .PAGE_WIDTH, ! cols to fill
1384 2317 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1385 2318 2
1386 2319 2 IF .RET_LEN[0] GTR 0
1387 2320 2 THEN
1388 2321 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
1389 2322 2 ! offset before inserting
1390 2323 2 ! includes the space
1391 2324 2 ! Insert the string delimited. Bottom of page.
1392 2325 2 INSERT_FRAME (.SCB,
1393 2326 2 STRING_DESC[0], ! string ref.
1394 2327 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], ! ref to frame
1395 2328 2 .PAGE_WIDTH, ! cols to fill
1396 2329 2 .RET_LEN[0]); ! rows to fill
1397 2330 2
1398 2331 2
1399 2332 2 ! Create a sentence describing the current file. Bottom of page.
1400 2333 2
1401 2334 2 ! re-init
1402 2335 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1403 2336 2
1404 2337 2 GET_FILE_DESCRIPTION
1405 2338 2 (.SCB, ! SCB addr.
1406 2339 2 STRING_DESC[0], ! Buffer descriptor
1407 2340 2 STRING_DESC[SIZE]); ! Returned length
1408 2341 2
1409 2342 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1410 2343 2 (.SCB,
1411 2344 2 STRING_DESC[0], ! string ref.
1412 2345 2 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
1413 2346 2 .PAGE_WIDTH, ! cols to fill
1414 2347 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1415 2348 2
1416 2349 2 IF .RET_LEN[0] GTR 0
1417 2350 2 THEN
1418 2351 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
1419 2352 2 ! offset before inserting
1420 2353 2
1421 2354 2 ! insert the string delimited
1422 2355 2 INSERT_FRAME (.SCB,
1423 2356 2 STRING_DESC[0], ! string ref.
1424 2357 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], ! ref to frame
1425 2358 2 .PAGE_WIDTH, ! cols to fill
1426 2359 2 .RET_LEN[0]); ! rows to fill
1427 2360 2
1428 2361 2
1429 2362 2 ! User note
1430 2363 2
1431 2364 2 ! re-init
1432 2365 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1433 2366 2
```

```
1434 2367 2 ! Get the user note
1435 2368 2 GET_USER_NOTE
1436 2369 2 (.SCB, ! SCB addr.
1437 2370 2 STRING_DESC[0], ! Buffer descriptor
1438 2371 2 STRING_DESC[SIZE]); ! Returned length
1439 2372 2
1440 2373 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1441 2374 2 (.SCB,
1442 2375 2 STRING_DESC[0], ! string ref.
1443 2376 2 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
1444 2377 2 .PAGE_WIDTH, ! cols to fill
1445 2378 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1446 2379 2
1447 2380 2 ! insert the string delimited
1448 2381 2 INSERT_FRAME (.SCB,
1449 2382 2 STRING_DESC[0], ! string ref.
1450 2383 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1451 2384 2 .PAGE_WIDTH, ! cols to fill
1452 2385 2 .RET_LEN[0]); ! rows to fill
1453 2386 2
1454 2387 2
1455 2388 2 ! User name
1456 2389 2
1457 2390 2 ! re-init
1458 2391 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1459 2392 2 IF .RET_LEN[0] GTR 0
1460 2393 2 THEN
1461 2394 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 1
1462 2395 2 ELSE
1463 2396 2 TOP_OFFSET = .TOP_OFFSET + 1;
1464 2397 2 ! adjust & allow for spacing
1465 2398 2
1466 2399 2 RET_LEN[0] = INSERT_NAME_BANNER (.SCB,
1467 2400 2 SCB_SIZE (USER NAME), ! user name desc
1468 2401 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1469 2402 2 .PAGE_WIDTH, ! max width Bann
1470 2403 2 .BOTTOM_OFFSET - .TOP_OFFSET, ! frame length
1471 2404 2 7); ! max hgt Bann str desired
1472 2405 2
1473 2406 2
1474 2407 2
1475 2408 2 IF .RET_LEN[0] GTR 0
1476 2409 2 THEN
1477 2410 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
1478 2411 2 ! adjust & allow for spacing
1479 2412 2
1480 2413 2 ! Get and insert the filename banner
1481 2414 2
1482 2415 2 ! re-init
1483 2416 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1484 2417 2 RET_LEN[0] = INSERT_FILENAME_BANNER
1485 2418 2 (.SCB,
1486 2419 2 STRING_DESC[0], ! Buffer desc.
1487 2420 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1488 2421 2 .PAGE_WIDTH, ! max width Bann
1489 2422 2 .BOTTOM_OFFSET - .TOP_OFFSET);
1490 2423 2
```

```
: 1491      2424  2      ! rows to fill
: 1492      2425  2      IF .RET_LEN[0] GTR 0
: 1493      2426  2      THEN
: 1494      2427  2      TOP_OFFSET      = .TOP_OFFSET + .RET_LEN[0] + 2;
: 1495      2428  2      ! adjust & allow for spacing
: 1496      2429  2
: 1497      2430  1 END;
```

```
OFFC 00000 FILL_FILE_FLAG:
      5B      0000V CF 9E 00002      .WORD      Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      2219
      5A      0000V CF 9E 00007      MOVAB      INSERT_FRAME, R11
      59      0000V CF 9E 0000C      MOVAB      RETURN_FRAME_LENGTH, R10
      5E      FDFC CE 9E 00011      MOVAB      FILL_FRAME, R9
      7E      0200 8F 3C 00016      MOVAB      -516(SP), SP
      04      AE      08 AE 9E 00018      MOVZWL     #512, STRING_DESC      2238
      53      10      AC      02 D4 00020      MOVAB      BUFFER, STRING_DESC+4
      57      0C      AC D0 00029      CLRL      TOP_OFFSET      2241
      54      08      AC D0 0002F      SUBL3     #2, PAGE_LENGTH, BOTTOM_OFFSET      2242
      52      57      C5 00033      PUSHL     #3
      56      04      AC D0 0003A      PUSHL     PAGE_WIDTH, R7      2249
      7E      02A4 C6 9A 0003E      PUSHL     R7
      69      05      FB 00045      MOVL      PAGE_REF, R4
      55      53      57      C5 0004C      MULL3     R7, TOP_OFFSET, R8
      7E      02A4 C6 9A 00053      PUSHAB    (R8)[R4]
      69      05      FB 0005A      MOVZBL    SCB, R6      2248
      EC      A7      9F 0005F      MOVZBL    676(R6), -(SP)
      0A      A844 9F 00062      PUSHL     R6      2249
      20      DD 00066      CALLS     #5, FILL_FRAME
      56      DD 00068      PUSHL     #3
      69      05      FB 0006A      PUSHL     R7
      EC      A7      9F 0006F      PUSHL     R7, BOTTOM_OFFSET, R5
      0A      A544 9F 00072      PUSHAB    (R5)[R4]
      20      DD 00076      MOVZBL    676(R6), -(SP)
      56      DD 00078      PUSHL     R6
      69      05      FB 0007A      CALLS     #5, FILL_FRAME
      EC      A7      9F 0007F      PUSHL     #3
      0A      A844 9F 00082      PUSHAB    -20(R7)
      20      DD 00076      PUSHAB    10(R8)[R4]
      56      DD 00078      PUSHL     #32
      69      05      FB 0007A      PUSHL     R6
      EC      A7      9F 0007F      CALLS     #5, FILL_FRAME
      0A      A844 9F 00082      PUSHAB    -20(R7)
      20      DD 00076      PUSHAB    10(R5)[R4]
      56      DD 00078      PUSHL     #32
      69      05      FB 0007A      PUSHL     R6
      EC      A7      9F 0007F      CALLS     #5, FILL_FRAME
      0A      A844 9F 00082      PUSHAB    -28(R7)
      20      DD 00076      PUSHAB    14(R8)[R4]
      56      DD 00078      MOVZBL    678(R6), -(SP)
      69      05      FB 0007A      PUSHL     R6
      EC      A7      9F 0007F      CALLS     #5, FILL_FRAME
      0A      A844 9F 00082      PUSHAB    14(R8)[R4]
      20      DD 00076      MOVZBL    678(R6), -(SP)
      56      DD 00078      PUSHL     R6
      69      05      FB 0007A      CALLS     #5, FILL_FRAME
```


			03	DD	00090	PUSHL	#3	2269
		F4	A7	9F	00092	PUSHAB	-28(R7)	
		OE	A544	9F	00095	PUSHAB	14(R5)[R4]	
	7E	02A6	C6	9A	00099	MOVZBL	678(R6), -(SP)	
			56	DD	0009E	PUSHL	R6	
	69		05	FB	000A0	CALLS	#5, FILL_FRAME	
	6E	0200	8F	3C	000A3	MOVZWL	#512, STRING_DESC	2276
			5E	DD	000A8	PUSHL	SP	2281
		04	AE	9F	000AA	PUSHAB	STRING_DESC	2280
			56	DD	000AD	PUSHL	R6	2279
0000V	CF		03	FB	000AF	CALLS	#3, GET_SYSTEM_ANNOUNCEMENT	
			01	DD	000B4	PUSHL	#1	2285
			57	DD	000B6	PUSHL	R7	
	58	01	A2	9E	000B8	MOVAB	1(R2), R8	
	58		57	C4	000BC	MULL2	R7, R8	
			6844	9F	000BF	PUSHAB	(R8)[R4]	
		0C	AE	9F	000C2	PUSHAB	STRING_DESC	2284
			56	DD	000C5	PUSHL	R6	2285
0000V	CF		05	FB	000C7	CALLS	#5, CENTER_FRAME	
	6E	0200	8F	3C	000CC	MOVZWL	#512, STRING_DESC	2288
			5E	DD	000D1	PUSHL	SP	2293
		04	AE	9F	000D3	PUSHAB	STRING_DESC	2292
			56	DD	000D6	PUSHL	R6	2291
0000V	CF		03	FB	000D8	CALLS	#3, GET_DIGITAL_LOGO	
			01	DD	000DD	PUSHL	#1	2297
			57	DD	000DF	PUSHL	R7	
	55	01	A3	9E	000E1	MOVAB	1(R3), R5	
	55		57	C4	000E5	MULL2	R7, R5	
			6544	9F	000E8	PUSHAB	(R5)[R4]	
		0C	AE	9F	000EB	PUSHAB	STRING_DESC	2296
			56	DD	000EE	PUSHL	R6	2297
0000V	CF		05	FB	000F0	CALLS	#5, CENTER_FRAME	
	6E	0200	8F	3C	000F5	MOVZWL	#512, STRING_DESC	2303
	52		04	C0	000FA	ADDL2	#4, TOP_OFFSET	2304
			5E	DD	000FD	PUSHL	SP	2310
		04	AE	9F	000FF	PUSHAB	STRING_DESC	2309
			01	DD	00102	PUSHL	#1	2307
			56	DD	00104	PUSHL	R6	
	0000V	CF	04	FB	00106	CALLS	#4, GET_JOB_DESCRIPTION	
7E	53		52	C3	00108	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2317
		0090	8F	BB	0010F	PUSHR	#*M2R4, R7>	2315
		0C	AE	9F	00113	PUSHAB	STRING_DESC	2314
			56	DD	00116	PUSHL	R6	2315
			05	FB	00118	CALLS	#5, RETURN_FRAME_LENGTH	
	6A		50	DD	0011B	MOVL	R0, RET_LEN	
	55		08	15	0011E	BLEQ	15	2319
50	53		55	C3	00120	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	2321
	53	FF	A0	9E	00124	MOVAB	-1(R0), BOTTOM_OFFSET	
			55	DD	00128	PUSHL	RET_LEN	2330
			57	DD	0012A	PUSHL	R7	2329
50	53		57	C3	0012C	MULL3	R7, BOTTOM_OFFSET, R0	2327
			6044	9F	00130	PUSHAB	(R0)[R4]	
		0C	AE	9F	00133	PUSHAB	STRING_DESC	2326
			56	DD	00136	PUSHL	R6	2327
	68		05	FB	00138	CALLS	#5, INSERT_FRAME	
	6E	0200	8F	3C	0013B	MOVZWL	#512, STRING_DESC	2335
			5E	DD	00140	PUSHL	SP	2340

			04	AE	9F	00142	PUSHAB	STRING_DESC	2339
				56	DD	00145	PUSHL	R6	2338
7E	0000V	CF		03	FB	00147	CALLS	#3, GET FILE DESCRIPTION	
		53		52	C3	0014C	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2347
			0090	8F	BB	00150	PUSHR	#^MZR4,R7>	2345
			0C	AE	9F	00154	PUSHAB	STRING_DESC	2344
				56	DD	00157	PUSHL	R6	2345
		6A		05	FB	00159	CALLS	#5, RETURN FRAME_LENGTH	
		55		50	DD	0015C	MOVL	R0, RET_LEN	
				08	15	0015F	BLEQ	2\$	2349
50		53		55	C3	00161	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	2351
		53	FF	A0	9E	00165	MOVAB	-1(R0), BOTTOM_OFFSET	
				55	DD	00169	PUSHL	RET_LEN	2360
				57	DD	0016B	PUSHL	R7	2359
50		53		57	C5	0016D	MULL3	R7, BOTTOM_OFFSET, R0	2357
			6044	9F	00171	PUSHAB	(R0)[R4]		
			0C	AE	9F	00174	PUSHAB	STRING_DESC	2356
				56	DD	00177	PUSHL	R6	2357
		6B		05	FB	00179	CALLS	#5, INSERT FRAME	
		6E	0200	8F	3C	0017C	MOVZWL	#512, STRING_DESC	2365
				5E	DD	00181	PUSHL	SP	2371
			04	AE	9F	00183	PUSHAB	STRING_DESC	2370
				56	DD	00186	PUSHL	R6	2369
7E	0000V	CF		03	FB	00188	CALLS	#3, GET USER NOTE	
		53		52	C3	0018D	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2378
			0090	8F	BB	00191	PUSHR	#^MZR4,R7>	2376
			0C	AE	9F	00195	PUSHAB	STRING_DESC	2375
				56	DD	00198	PUSHL	R6	2376
		6A		05	FB	0019A	CALLS	#5, RETURN FRAME_LENGTH	
		55		50	DD	0019D	MOVL	R0, RET_LEN	
				55	DD	001A0	PUSHL	RET_LEN	2386
				57	DD	001A2	PUSHL	R7	2385
58		52		57	C5	001A4	MULL3	R7, TOP_OFFSET, R8	2383
			6844	9F	001A8	PUSHAB	(R8)[R4]		
			0C	AE	9F	001AB	PUSHAB	STRING_DESC	2382
				56	DD	001AE	PUSHL	R6	2383
		6B		05	FB	001B0	CALLS	#5, INSERT FRAME	
		6E	0200	8F	3C	001B3	MOVZWL	#512, STRING_DESC	2391
				55	D5	001B8	TSTL	RET_LEN	2392
				07	15	001BA	BLEQ	3\$	
		52	01	A542	9E	001BC	MOVAB	1(RET_LEN)[TOP_OFFSET], TOP_OFFSET	2394
				02	11	001C1	BRB	4\$	
				52	D6	001C3	INCL	TOP_OFFSET	2396
				07	DD	001C5	PUSHL	#7	2401
7E		53		52	C3	001C7	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2404
				57	DD	001CB	PUSHL	R7	2403
58		52		57	C5	001CD	MULL3	R7, TOP_OFFSET, R8	2401
			6844	9F	001D1	PUSHAB	(R8)[R4]		
			016C	C6	9F	001D4	PUSHAB	364(R6)	2400
				56	DD	001D8	PUSHL	R6	2401
	0000V	CF		06	FB	001DA	CALLS	#6, INSERT NAME_BANNER	
		55		50	DD	001DF	MOVL	R0, RET_LEN	
				05	15	001E2	BLEQ	5\$	2408
		52	02	A542	9E	001E4	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	2410
		6E	0200	8F	3C	001E9	MOVZWL	#512, STRING_DESC	2416
7E		53		52	C3	001EE	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2423
				57	DD	001F2	PUSHL	R7	2422

SEPARATE
V04-001

Print Symbiont -- separation routines

FILL_FILE_FLAG - Insert Information into the FI

L 16

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 47
(15)

58

52

57 C5 001F4
6844 9F 001F8
OC AE 9F 001FB
56 DD 001FE
05 FB 00200
50 D0 00205
05 15 00208
52 02 A542 9E 0020A
04 0020F 68:

0000V

CF

55

MULL3 R7, TOP_OFFSET, R8
PUSHAB (R8)[R4]
PUSHAB STRING_DESC
PUSHL R6
CALLS #5, INSERT_FILENAME_BANNER
MOVL R0, RET_LEN
BLEQ 68
MOVAB 2(RET_LEN)[TOP_OFFSET], TOP_OFFSET
RET

: 2420
: 2419
: 2420
: 2425
: 2427
: 2430

: Routine Size: 528 bytes, Routine Base: CODE + 077F

```
1499 2431 1 %sbttl 'FILL_JOB_FLAG - Insert Information into the JOB Page'
1500 2432 1 ++
1501 2433 1 Functional Description:
1502 2434 1 This procedure controls all inserts required for the JOB Page.
1503 2435 1
1504 2436 1 Formal Parameters:
1505 2437 1 SCB - Address of the SCB
1506 2438 1 PAGE_REF - Pointer to the Page (first byte)
1507 2439 1 PAGE_LENGTH - Length of Frame
1508 2440 1 PAGE_WIDTH - Width of Frame
1509 2441 1
1510 2442 1 Implicit Inputs:
1511 2443 1 none
1512 2444 1
1513 2445 1 Implicit Outputs:
1514 2446 1 none
1515 2447 1
1516 2448 1 Returned Value:
1517 2449 1 none
1518 2450 1
1519 2451 1 Side Effects:
1520 2452 1 none
1521 2453 1 --
1522 2454 1 ROUTINE FILL_JOB_FLAG (
1523 2455 1 SCB : REF $BLOCK,
1524 2456 1 PAGE_REF : REF PAGE_ARRAY,
1525 2457 1 PAGE_WIDTH,
1526 2458 1 PAGE_LENGTH
1527 2459 1 ): NOVALUE =
1528 2460 2 BEGIN
1529 2461 2
1530 2462 2 LITERAL K_MAX_BUFFER_SIZE = 512;
1531 2463 2
1532 2464 2 LOCAL
1533 2465 2 RET_LEN : VECTOR[1],
1534 2466 2 TOP_OFFSET
1535 2467 2 BOTTOM_OFFSET,
1536 2468 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
1537 2469 2 STRING_DESC : VECTOR [2]; ! Descriptor to current string
1538 2470 2
1539 2471 2 ! Allocate the buffer for "GET_xxx" Routines
1540 2472 2
1541 2473 2 STRING_DESC[SIZE] = %ALLOCATION(BUFFER); ! allocate for routines
1542 2474 2 STRING_DESC[ADDR] = BUFFER; ! init address
1543 2475 2
1544 2476 2 TOP_OFFSET = 0;
1545 2477 2 BOTTOM_OFFSET = .PAGE_LENGTH - 2; ! offset includes burst offset
1546 2478 2
1547 2479 2 ! Burst Character
1548 2480 2 !
1549 2481 2 FILL_FRAME (.SCB,
1550 2482 2 .SCB[PSM$B_JOB BURST CHAR],
1551 2483 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
1552 2484 2
1553 2485 2 FILL_FRAME (.SCB,
1554 2486 2 .SCB[PSM$B_JOB BURST CHAR],
1555 2487 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
```



```
1556 2488 2
1557 2489 2
1558 2490 2
1559 2491 2
1560 2492 2
1561 2493 2
1562 2494 2
1563 2495 2
1564 2496 2
1565 2497 2
1566 2498 2
1567 2499 2
1568 2500 2
1569 2501 2
1570 2502 2
1571 2503 2
1572 2504 2
1573 2505 2
1574 2506 2
1575 2507 2
1576 2508 2
1577 2509 2
1578 2510 2
1579 2511 2
1580 2512 2
1581 2513 2
1582 2514 2
1583 2515 2
1584 2516 2
1585 2517 2
1586 2518 2
1587 2519 2
1588 2520 2
1589 2521 2
1590 2522 2
1591 2523 2
1592 2524 2
1593 2525 2
1594 2526 2
1595 2527 2
1596 2528 2
1597 2529 2
1598 2530 2
1599 2531 2
1600 2532 2
1601 2533 2
1602 2534 2
1603 2535 2
1604 2536 2
1605 2537 2
1606 2538 2
1607 2539 2
1608 2540 2
1609 2541 2
1610 2542 2
1611 2543 2
1612 2544 2

! System announcement
! re-init
STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
GET_SYSTEM_ANNOUNCEMENT
      (.SCB,
        STRING_DESC[0],
        STRING_DESC[SIZE]);
      ! SCB addr.
      ! Buffer descriptor
      ! Returned length

CENTER_FRAME (.SCB,
              STRING_DESC[0],
              PAGE_REF[0,.TOP_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);

! re-init
STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
GET_DIGITAL_LOGO
      (.SCB,
        STRING_DESC[0],
        STRING_DESC[SIZE]);
      ! SCB addr.
      ! Buffer descriptor
      ! Returned length

CENTER_FRAME (.SCB,
              STRING_DESC[0],
              PAGE_REF[0,.BOTTOM_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);

TOP_OFFSET = .TOP_OFFSET + 4;                ! adjust & allow for spacing

! Job description - create a sentence describing the current job.
! re-init
STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
GET_JOB_DESCRIPTION
      (.SCB,
        !
        STRING_DESC[0],
        STRING_DESC[SIZE]);
      ! SCB addr.
      ! Use present tense
      ! Buffer descriptor
      ! Returned length

RET_LEN[0] = RETURN_FRAME_LENGTH
      (.SCB,
        STRING_DESC[0],
        PAGE_REF[0,0,.PAGE_WIDTH],
        .PAGE_WIDTH,
        .BOTTOM_OFFSET - .TOP_OFFSET);
      ! string ref.
      ! ref to frame
      ! cols to fill
      ! rows to fill

IF .RET_LEN[0] GTR 0
THEN
  BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
  ! offset before inserting
! insert the string delimited
INSERT_FRAME (.SCB,
              STRING_DESC[0],
              PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH],
              .PAGE_WIDTH,
              .RET_LEN[0]);
              ! string ref.
              ! ref to frame
              ! cols to fill
              ! rows to fill

! User note
```

```
1613 2545 2 !
1614 2546 ! re-init
1615 2547 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1616 2548
1617 2549 ! Get the user note
1618 2550 GET_USER_NOTE
1619 2551 (.SCB, ! SCB addr.
1620 2552 STRING_DESC[0], ! Buffer descriptor
1621 2553 STRING_DESC[SIZE]); ! Returned length
1622 2554
1623 2555 RET_LEN[0] = RETURN_FRAME_LENGTH
1624 2556 (.SCB,
1625 2557 STRING_DESC[0], ! string ref.
1626 2558 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
1627 2559 .PAGE_WIDTH, ! cols to fill
1628 2560 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1629 2561
1630 2562 ! insert the string delimited
1631 2563 INSERT_FRAME (.SCB,
1632 2564 STRING_DESC[0], ! string ref.
1633 2565 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1634 2566 .PAGE_WIDTH, ! cols to fill
1635 2567 .RET_LEN[0]); ! rows to fill
1636 2568
1637 2569
1638 2570 IF .RET_LEN[0] GTR 0
1639 2571 THEN
1640 2572 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 1
1641 2573 ELSE
1642 2574 TOP_OFFSET = .TOP_OFFSET + 1;
1643 2575 ! adjust & allow for spacing
1644 2576
1645 2577 ! User Name
1646 2578 ! re-init
1647 2579 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1648 2580
1649 2581 RET_LEN[0] = INSERT_NAME_BANNER (
1650 2582 .SCB,
1651 2583 SCB_SIZE (USER_NAME), ! user name descriptor
1652 2584 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1653 2585 .PAGE_WIDTH, ! max width Bann
1654 2586 .BOTTOM_OFFSET - .TOP_OFFSET,
1655 2587 ! space left
1656 2588 14); ! max hgt Bann string desired
1657 2589
1658 2590
1659 2591 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2; ! adjust for banner & spacing
1660 2592
1661 2593 ! Job Name
1662 2594 ! re-init
1663 2595 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1664 2596
1665 2597 GET_JOB_NAME
1666 2598 (.SCB, ! SCB addr.
1667 2599 STRING_DESC[0], ! Buffer descriptor
1668 2600 STRING_DESC[SIZE]); ! Returned length
1669 2601
```

```
1670 2602 2
1671 2603 2 RET_LEN[0] = INSERT_NAME_BANNER (
1672 2604 2 .SCB,
1673 2605 2 STRING_DESC[SIZE], ! job name desc
1674 2606 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH],
1675 2607 2 ! ref to frame
1676 2608 2 .PAGE_WIDTH, ! max width Bann
1677 2609 2 .BOTTOM_OFFSET-.TOP_OFFSET,
1678 2610 2 ! space left
1679 2611 2 7); ! max hght Bann str desired
1680 2612 2
1681 2613 2 IF .RET_LEN[0] GTR 0
1682 2614 2 THEN
1683 2615 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
1684 2616 2 ! adjust & allow for spacing
1685 2617 2 ! Get and insert the filename banner
1686 2618 2 !
1687 2619 2
1688 2620 2 ! re-init
1689 2621 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1690 2622 2
1691 2623 2 IF (.BOTTOM_OFFSET - 9) GTR .TOP_OFFSET ! test for enough room
1692 2624 2 THEN
1693 2625 2 BEGIN
1694 2626 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - 9;
1695 2627 2 ! offset before inserting
1696 2628 2 INSERT_JOBNUMBER_BANNER
1697 2629 2 (.SCB,
1698 2630 2 STRING_DESC[0], ! Buffer desc.
1699 2631 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH],
1700 2632 2 ! ref to frame
1701 2633 2 .PAGE_WIDTH, ! max width Bann
1702 2634 2 7); ! rows to fill
1703 2635 2 END;
1704 2636 2 1 END;
```

00FC 0000 FILL_JOB_FLAG:

	5E	FDFC	CE	9E	00002	WORD	Save R2,R3,R4,R5,R6,R7	2454
	7E	0200	8F	3C	00007	MOVAB	-516(SP), SP	
	04	AE	AE	9E	0000C	MOVZWL	#512, STRING_DESC	2473
			52	D4	00011	MOVAB	BUFFER, STRING_DESC+4	2474
53	10	AC	02	C3	00013	CLRL	TOP_OFFSET	2476
			03	DD	00018	SUBL3	#2, PAGE_LENGTH, BOTTOM_OFFSET	2477
	57	0C	AC	D0	0001A	PUSHL	#3	2483
			57	DD	0001E	MOVL	PAGE_WIDTH, R7	
	54	08	AC	D0	00020	PUSHL	R7	
50	52		57	C5	00024	MOVL	PAGE_REF, R4	
			6044	9F	00028	MULL3	R7, TOP_OFFSET, R0	
	56	04	AC	D0	0002B	PUSHAB	(R0)[R4]	
	7E	02A6	C6	9A	0002F	MOVL	SCB, R6	2482
			56	DD	00034	MOVZBL	678(R6), -(SP)	2483
0000V	CF		05	FB	00036	PUSHL	R6	
						CALLS	#5, FILL_FRAME	

50	53		03 DD 0003B	PUSHL #3	2487
			57 DD 0003D	PUSHL R7	
	7E	02A6	57 C5 0003F	MULL3 R7, BOTTOM_OFFSET, R0	
			6044 9F 00043	PUSHAB (R0)[R4]	
			C6 9A 00046	MOVZBL 678(R6), -(SP)	
			56 DD 0004B	PUSHL R6	
0000V	CF		05 FB 0004D	CALLS #5, FILL_FRAME	
	6E	0200	8F 3C 00052	MOVZWL #512, STRING_DESC	2492
			5E DD 00057	PUSHL SP	2496
		04	AE 9F 00059	PUSHAB STRING_DESC	2495
			56 DD 0005C	PUSHL R6	2494
0000V	CF		03 FB 0005E	CALLS #3, GET_SYSTEM_ANNOUNCEMENT	
			01 DD 00063	PUSHL #1	2500
			57 DD 00065	PUSHL R7	
	50	01	A2 9E 00067	MOVAB 1(R2), R0	
	50		57 C4 0006B	MULL2 R7, R0	
			6044 9F 0006E	PUSHAB (R0)[R4]	
		0C	AE 9F 00071	PUSHAB STRING_DESC	2499
			56 DD 00074	PUSHL R6	2500
0000V	CF		05 FB 00076	CALLS #5, CENTER_FRAME	
	6E	0200	8F 3C 0007B	MOVZWL #512, STRING_DESC	2503
			5E DD 00080	PUSHL SP	2507
		04	AE 9F 00082	PUSHAB STRING_DESC	2506
			56 DD 00085	PUSHL R6	2505
0000V	CF		03 FB 00087	CALLS #3, GET_DIGITAL_LOGO	
			01 DD 0008C	PUSHL #1	2511
			57 DD 0008E	PUSHL R7	
	50	01	A3 9E 00090	MOVAB 1(R3), R0	
	50		57 C4 00094	MULL2 R7, R0	
			6044 9F 00097	PUSHAB (R0)[R4]	
		0C	AE 9F 0009A	PUSHAB STRING_DESC	2510
			56 DD 0009D	PUSHL R6	2511
0000V	CF		05 FB 0009F	CALLS #5, CENTER_FRAME	
	52		04 C0 000A4	ADDL2 #4, TOP_OFFSET	2513
	6E	0200	8F 3C 000A7	MOVZWL #512, STRING_DESC	2518
			5E DD 000AC	PUSHL SP	2523
		04	AE 9F 000AE	PUSHAB STRING_DESC	2522
			01 DD 000B1	PUSHL #1	2520
			56 DD 000B3	PUSHL R6	
0000V	CF		04 FB 000B5	CALLS #4, GET_JOB_DESCRIPTION	
	53		52 C3 000BA	SUBL3 TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2530
		0090	8F BB 000BE	PUSHR #4, R7	2528
		0C	AE 9F 000C2	PUSHAB STRING_DESC	2527
			56 DD 000C5	PUSHL R6	2528
0000V	CF		05 FB 000C7	CALLS #5, RETURN_FRAME_LENGTH	
	55		50 D0 000CC	MOVL R0, RET_LEN	
			08 15 000CF	BLEQ 15	2532
50	53		55 C3 000D1	SUBL3 RET_LEN, BOTTOM_OFFSET, R0	2534
	53	FF	A0 9E 000D5	MOVAB -1(R0), BOTTOM_OFFSET	
			55 DD 000D9	PUSHL RET_LEN	2542
			57 DD 000DB	PUSHL R7	2541
50	53		57 C5 000DD	MULL3 R7, BOTTOM_OFFSET, R0	2539
			6044 9F 000E1	PUSHAB (R0)[R4]	
		0C	AE 9F 000E4	PUSHAB STRING_DESC	2538
			56 DD 000E7	PUSHL R6	2539
0000V	CF		05 FB 000E9	CALLS #5, INSERT_FRAME	
	6E	0200	8F 3C 000EE	MOVZWL #512, STRING_DESC	2547

18:

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_FLAG - Insert Information into the JOB

F 1

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 B11ss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 53
(16)

			04	5E DD 000F3	PUSHL SP	2553
				AE 9F 000F5	PUSHAB STRING_DESC	2552
				56 DD 000F8	PUSHL R6	2551
7E	0000V	CF		03 FB 000FA	CALLS #3, GET_USER_NOTE	
		53		52 C3 000FF	SUBL3 TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2560
			0090	8F BB 00103	PUSHR #MZR4, R7>	2558
			0C	AE 9F 00107	PUSHAB STRING_DESC	2557
				56 DD 0010A	PUSHL R6	2558
	0000V	CF		05 FB 0010C	CALLS #5, RETURN_FRAME_LENGTH	
		55		50 D0 00111	MOVL R0, RET_LEN	
				55 DD 00114	PUSHL RET_LEN	2568
50		52		57 DD 00116	PUSHL R7	2567
				57 C5 00118	MULL3 R7, TOP_OFFSET, R0	2565
			6044	9F 0011C	PUSHAB (R0)[R4]	
			0C	AE 9F 0011F	PUSHAB STRING_DESC	2564
				56 DD 00122	PUSHL R6	2565
	0000V	CF		05 FB 00124	CALLS #5, INSERT_FRAME	
				55 D5 00129	TSTL RET_LEN	2570
				07 15 0012B	BLEQ 2\$	
		52	01	A542 9E 0012D	MOVAB 1(RET_LEN)[TOP_OFFSET], TOP_OFFSET	2572
				02 11 00132	BRB 3\$	
				52 D6 00134	INCL TOP_OFFSET	2574
		6E	0200	8F 3C 00136	MOVZWL #512, STRING_DESC	2579
				0E DD 0013B	PUSHL #14	2584
7E		53		52 C3 0013D	SUBL3 TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2587
				57 DD 00141	PUSHL R7	2586
50		52		57 C5 00143	MULL3 R7, TOP_OFFSET, R0	2584
			6044	9F 00147	PUSHAB (R0)[R4]	
			016C	C6 9F 0014A	PUSHAB 364(R6)	2583
				56 DD 0014E	PUSHL R6	2584
	0000V	CF		06 FB 00150	CALLS #6, INSERT_NAME_BANNER	
		55		50 D0 00155	MOVL R0, RET_LEN	
		52	02	A542 9E 00158	MOVAB 2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	2591
		6E	0200	8F 3C 0015D	MOVZWL #512, STRING_DESC	2596
				5E DD 00162	PUSHL SP	2601
			04	AE 9F 00164	PUSHAB STRING_DESC	2600
				56 DD 00167	PUSHL R6	2599
	0000V	CF		03 FB 00169	CALLS #3, GET_JOB_NAME	
				07 DD 0016E	PUSHL #7	2606
7E		53		52 C3 00170	SUBL3 TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2609
				57 DD 00174	PUSHL R7	2608
50		52		57 C5 00176	MULL3 R7, TOP_OFFSET, R0	2606
			6044	9F 0017A	PUSHAB (R0)[R4]	
			10	AE 9F 0017D	PUSHAB STRING_DESC	2605
				56 DD 00180	PUSHL R6	2606
	0000V	CF		06 FB 00182	CALLS #6, INSERT_NAME_BANNER	
		55		50 D0 00187	MOVL R0, RET_LEN	
				05 15 0018A	BLEQ 4\$	2613
		52	02	A542 9E 0018C	MOVAB 2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	2615
		6E	0200	8F 3C 00191	MOVZWL #512, STRING_DESC	2621
		50	F7	A3 9E 00196	MOVAB -9(R3), R0	2623
		52		50 D1 0019A	CMPL R0, TOP_OFFSET	
				17 15 0019D	BLEQ 5\$	
		53		09 C2 0019F	SUBL2 #9, BOTTOM_OFFSET	2626
				07 DD 001A2	PUSHL #7	2631
				57 DD 001A4	PUSHL R7	2633
		53		57 C4 001A6	MULL2 R7, R3	2631

SEP
V04

SEPARATE
V04-001

Print Symbiont -- separation routines

FILL_JOB_FLAG - Insert Information into the JOB

G 1
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 54
(16)

	6344	9F	001A9	PUSHAB	(R3)[R4]
OC	AE	9F	001AC	PUSHAB	STRING_DESC
	56	DD	001AF	PUSHL	R6
0000V CF	05	FB	001B1	CALLS	#5, INSERT_JOBNUMBER_BANNER
	04	001B6	5\$:	RET	

: 2630
: 2631
: 2636

; Routine Size: 439 bytes. Routine Base: CODE + 098F

```
1706 2637 1 %sbttl 'FILL_JOB_TRAILER - Insert Information into the JOB Page'
1707 2638 1 ++
1708 2639 1 Functional Description:
1709 2640 1 This procedure controls all inserts required for the JOB Page.
1710 2641 1
1711 2642 1 Formal Parameters:
1712 2643 1 SCB - Address of the SCB
1713 2644 1 PAGE_REF - Pointer to the Page (first byte)
1714 2645 1 PAGE_LENGTH - Length of Frame
1715 2646 1 PAGE_WIDTH - Width of Frame
1716 2647 1
1717 2648 1 Implicit Inputs:
1718 2649 1 none
1719 2650 1
1720 2651 1 Implicit Outputs:
1721 2652 1 none
1722 2653 1
1723 2654 1 Returned Value:
1724 2655 1 none
1725 2656 1
1726 2657 1 Side Effects:
1727 2658 1 none
1728 2659 1 --
1729 2660 1 ROUTINE FILL_JOB_TRAILER (
1730 2661 1 SCB : REF $BBLOCK,
1731 2662 1 PAGE_REF : REF PAGE_ARRAY,
1732 2663 1 PAGE_WIDTH,
1733 2664 1 PAGE_LENGTH
1734 2665 1 ): NOVALUE =
1735 2666 2 BEGIN
1736 2667 2
1737 2668 2 LITERAL K_MAX_BUFFER_SIZE = 512;
1738 2669 2
1739 2670 2 LOCAL
1740 2671 2 RET_LEN : VECTOR[1],
1741 2672 2 RET_WIDE : VECTOR[1],
1742 2673 2 RIGHT_OFFSET,
1743 2674 2 LEFT_OFFSET,
1744 2675 2 TOP_OFFSET,
1745 2676 2 BOTTOM_OFFSET,
1746 2677 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
1747 2678 2 STRING_DESC : VECTOR [2]; ! Descriptor to current string
1748 2679 2
1749 2680 2 ! Allocate the buffer for "GET_xxx" Routines
1750 2681 2
1751 2682 2 STRING_DESC[SIZE] = %ALLOCATION(BUFFER); ! allocate for routines
1752 2683 2 STRING_DESC[ADDR] = BUFFER; ! init address
1753 2684 2
1754 2685 2 ! Top of page
1755 2686 2
1756 2687 2 TOP_OFFSET = 0;
1757 2688 2 BOTTOM_OFFSET = .PAGE_LENGTH;
1758 2689 2
1759 2690 2 ! insert the burst characters
1760 2691 2 FILL_FRAME (.SCB,
1761 2692 2 .SCB[PSMSB_JOB BURST CHAR],
1762 2693 2 PAGE_REF[0, .TOP_OFFSET+2, .PAGE_WIDTH], .PAGE_WIDTH, 3);
```

```
1763 2694 2
1764 2695 2
1765 2696 2 ! re-init
1766 2697 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1767 2698 2
1768 2699 2 GET_EOJ
1769 2700 2 (.SCB,
1770 2701 2 STRING_DESC[0], ! Buffer descriptor
1771 2702 2 STRING_DESC[SIZE]); ! Returned length
1772 2703 2
1773 2704 2 RET_LEN[0] = INSERT_NAME_BANNER (
1774 2705 2 .SCB,
1775 2706 2 STRING_DESC[SIZE], ! eoJ name desc
1776 2707 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1777 2708 2 .PAGE_WIDTH, ! max width Bann
1778 2709 2 .BOTTOM_OFFSET - .TOP_OFFSET,
1779 2710 2 ! frame size
1780 2711 2 7); ! max hght Bann str
1781 2712 2
1782 2713 2 IF .RET_LEN[0] GTR 0
1783 2714 2 THEN
1784 2715 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
1785 2716 2 ! adjust & allow for spacing
1786 2717 2 ! includes the burst also...
1787 2718 2 ! two spaces...
1788 2719 2
1789 2720 2 ! Bottom of page
1790 2721 2 BOTTOM_OFFSET = .PAGE_LENGTH - 5; ! offset includes burst offset
1791 2722 2
1792 2723 2 FILL_FRAME (.SCB,
1793 2724 2 .SCB[PSMSB_JOB_BURST_CHAR],
1794 2725 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH,3);
1795 2726 2
1796 2727 2 ! re-init
1797 2728 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1798 2729 2
1799 2730 2 ! Get the digital logo and output to page ... assume not greater than
1800 2731 2 ! amount allocated.... truncation otherwise occurs
1801 2732 2
1802 2733 2 GET_DIGITAL_LOGO
1803 2734 2 (.SCB, ! SCB addr.
1804 2735 2 STRING_DESC[0], ! Buffer descriptor
1805 2736 2 STRING_DESC[SIZE]); ! Returned length
1806 2737 2
1807 2738 2 CENTER_FRAME (.SCB,
1808 2739 2 STRING_DESC[0],
1809 2740 2 PAGE_REF[0,.BOTTOM_OFFSET+1,.PAGE_WIDTH], .PAGE_WIDTH, 1);
1810 2741 2
1811 2742 2
1812 2743 2 ! re-init
1813 2744 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1814 2745 2
1815 2746 2 GET_RULER_COARSE
1816 2747 2 (.SCB, ! SCB addr.
1817 2748 2 STRING_DESC[0], ! Buffer descriptor
1818 2749 2 STRING_DESC[SIZE]); ! Returned length
1819 2750 2
```



```
1820 2751 2 SCROLL_FRAME (.SCB,  
1821 2752 2 STRING_DESC[0],  
1822 2753 2 PAGE_REF[0,.BOTTOM_OFFSET+4,.PAGE_WIDTH], .PAGE_WIDTH, 1);  
1823 2754 2  
1824 2755 2 SCROLL_FRAME (.SCB,  
1825 2756 2 $DESCRIPTOR ('1234567890'),  
1826 2757 2 PAGE_REF[0,.BOTTOM_OFFSET+5,.PAGE_WIDTH], .PAGE_WIDTH, 1);  
1827 2758 2  
1828 2759 2 ! Create a sentence describing the current job.  
1829 2760 2  
1830 2761 2 ! re-init  
1831 2762 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size  
1832 2763 2  
1833 2764 2 GET_JOB_DESCRIPTION  
1834 2765 2 (.SCB, ! SCB addr.  
1835 2766 2 0, ! Use past tense  
1836 2767 2 STRING_DESC[0], ! Buffer descriptor  
1837 2768 2 STRING_DESC[SIZE]); ! Returned length  
1838 2769 2  
1839 2770 2 RET_LEN[0] = RETURN_FRAME_LENGTH  
1840 2771 2 (.SCB,  
1841 2772 2 STRING_DESC[0], ! string ref.  
1842 2773 2 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame  
1843 2774 2 .PAGE_WIDTH, ! cols to fill  
1844 2775 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill  
1845 2776 2  
1846 2777 2 IF .RET_LEN[0] GTR 0  
1847 2778 2 THEN  
1848 2779 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);  
1849 2780 2 ! adjust & allow for spacing  
1850 2781 2 ! before inserting  
1851 2782 2 ! insert the string delimited  
1852 2783 2 INSERT_FRAME (.SCB,  
1853 2784 2 STRING_DESC[0], ! string ref.  
1854 2785 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], ! ref to frame  
1855 2786 2 .PAGE_WIDTH, ! cols to fill  
1856 2787 2 .RET_LEN[0]); ! rows to fill  
1857 2788 2  
1858 2789 2  
1859 2790 2  
1860 2791 2 ! User name  
1861 2792 2  
1862 2793 2 RET_LEN[0] = INSERT_NAME_BANNER  
1863 2794 2 (.SCB,  
1864 2795 2 SCB_SIZE (USER NAME), ! user name desc  
1865 2796 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame  
1866 2797 2 .PAGE_WIDTH, ! max width Bann  
1867 2798 2 .BOTTOM_OFFSET - .TOP_OFFSET, ! frame size  
1868 2799 2 7); ! max hght Bann str desired  
1869 2800 2  
1870 2801 2  
1871 2802 2 ! re-init  
1872 2803 2 IF .RET_LEN[0] GTR 0  
1873 2804 2 THEN  
1874 2805 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;  
1875 2806 2 ! adjust & allow for spacing  
1876 2807 2
```

```
1877 2808 2 ! Job name
1878 2809 2 !
1879 2810 2 ! re-init
1880 2811 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1881 2812 2
1882 2813 2 GET_JOB_NAME
1883 2814 2 (.SCB,
1884 2815 2 STRING_DESC[0], ! Buffer descriptor
1885 2816 2 STRING_DESC[SIZE]); ! Returned length
1886 2817 2
1887 2818 2 RET_LEN[0] = INSERT_NAME_BANNER (
1888 2819 2 .SCB,
1889 2820 2 STRING_DESC[SIZE], ! job name size
1890 2821 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1891 2822 2 .PAGE_WIDTH, ! max width Bann
1892 2823 2 .BOTTOM_OFFSET - .TOP_OFFSET, ! frame size
1893 2824 2 7); ! max hght Bann str
1894 2825 2
1895 2826 2
1896 2827 2 ! re-init
1897 2828 2 IF .RET_LEN[0] GTR 0
1898 2829 2 THEN
1899 2830 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
1900 2831 2 ! adjust & allow for spacing
1901 2832 2
1902 2833 2 ! Receipt box
1903 2834 2
1904 2835 2 RIGHT_OFFSET = .PAGE_WIDTH; ! right_offset and left_offset
1905 2836 2 LEFT_OFFSET = 0; ! are positional offsets for
1906 2837 2 ! a specific frame_length and
1907 2838 2 ! range.
1908 2839 2
1909 2840 2 ! re-init
1910 2841 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1911 2842 2
1912 2843 2 GET_RECEIPT_BOX
1913 2844 2 (.SCB, ! SCB addr.
1914 2845 2 STRING_DESC[0], ! Buffer descriptor
1915 2846 2 STRING_DESC[SIZE]); ! Returned length
1916 2847 2
1917 2848 2 ! Get the width needed for insert (assume length of seven)
1918 2849 2 RET_WIDE[0] = RETURN_FRAME_WIDTH
1919 2850 2 (.SCB,
1920 2851 2 STRING_DESC[0], ! string ref.
1921 2852 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1922 2853 2 .RIGHT_OFFSET - .LEFT_OFFSET, ! special width
1923 2854 2 8); ! rows to fill
1924 2855 2
1925 2856 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1926 2857 2 (.SCB,
1927 2858 2 STRING_DESC[0], ! string ref.
1928 2859 2 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
1929 2860 2 .RET_WIDE[0], ! cols to fill
1930 2861 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1931 2862 2
1932 2863 2 IF .RET_LEN[0] GTR 0
1933 2864 2 THEN
```

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information into the

L 1
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 59
(17)

```
1934 2865 2      BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
1935 2866 2      ! adjust & allow for spacing
1936 2867 2      ! before inserting
1937 2868 2      RIGHT_OFFSET = .RIGHT_OFFSET - .RET_WIDE[0]; ! offset before inserting
1938 2869 2
1939 2870 2      MOVE_FRAME (.SCB,
1940 2871 2      STRING_DESC[0], ! string frame reference
1941 2872 2      PAGE_REF[.RIGHT_OFFSET, .BOTTOM_OFFSET, .PAGE_WIDTH],
1942 2873 2      ! ref to frame
1943 2874 2      .RET_WIDE[0], ! width
1944 2875 2      .RET_LEN[0]); ! rows to fill
1945 2876 2
1946 2877 2      ! Get and insert the filename banner
1947 2878 2
1948 2879 2
1949 2880 2      ! re-init
1950 2881 2      STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1951 2882 2
1952 2883 2      INSERT_JOBNUMBER_BANNER
1953 2884 2      (.SCB,
1954 2885 2      STRING_DESC[0], ! Buffer desc.
1955 2886 2      PAGE_REF[.LEFT_OFFSET, .BOTTOM_OFFSET, .PAGE_WIDTH],
1956 2887 2      ! ref to frame
1957 2888 2      .RIGHT_OFFSET - .LEFT_OFFSET, ! max width Bann
1958 2889 2      .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1959 2890 2
1960 2891 2      ! re-init
1961 2892 2      STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1962 2893 2      RIGHT_OFFSET = .PAGE_WIDTH; ! right_offset and left_offset
1963 2894 2      LEFT_OFFSET = 0; ! are positional offsets for
1964 2895 2      ! a specific frame_length and
1965 2896 2      ! range.
1966 2897 2
1967 2898 2      GET_ACCOUNTING_INFO
1968 2899 2      (.SCB,
1969 2900 2      STRING_DESC[0], ! Buffer descriptor
1970 2901 2      STRING_DESC[SIZE]); ! Returned length
1971 2902 2
1972 2903 2      insert the string delimited
1973 2904 2      INSERT_FRAME (.SCB,
1974 2905 2      STRING_DESC[0], ! string ref.
1975 2906 2      PAGE_REF[0, 45, .PAGE_WIDTH], ! ref to frame
1976 2907 2      .PAGE_WIDTH, ! cols to fill
1977 2908 2      2); ! rows to fill
1978 2909 1      END;
```

```
30 39 38 37 36 35 34 33 32 31 00B46 P.AAF: .ASCII \1234567890\
0000000A 00B50 P.AAE: .LONG 10
00000000 00B54 .ADDRESS P.AAF
```

```
OFFC 00000 FILL_JOB_TRAILER:
WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
5B 0000V CF 9E 00002 MOVAB INSERT_NAME_BANNER, R11
```

: 2660
:

Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information into the

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 60
(17)

SEP
V04

	SE	FDFC	CE	9E	00007	MOVAB	-516(SP), SP	:	2682
	7E	0200	8F	3C	0000C	MOVZWL	#512, STRING_DESC	:	2683
04	AE	08	AE	9E	00011	MOVAB	BUFFER, STRING_DESC+4	:	2687
			52	D4	00016	CLRL	TOP_OFFSET	:	2688
53		10	AC	D0	00018	MOVL	PAGE_LENGTH, BOTTOM_OFFSET	:	2693
			03	DD	0001C	PUSHL	#3	:	
57		0C	AC	D0	0001E	MOVL	PAGE_WIDTH, R7	:	
			57	DD	00022	PUSHL	R7	:	
54		08	AC	D0	00024	MOVL	PAGE_REF, R4	:	
50		02	A2	9E	00028	MOVAB	2(R2), R0	:	
50			57	C4	0002C	MULL2	R7, R0	:	
			6044	9F	0002F	PUSHAB	(R0)[R4]	:	
56		04	AC	D0	00032	MOVL	SCB, R6	:	2692
7E		02A6	C6	9A	00036	MOVZBL	678(R6), -(SP)	:	2693
			56	DD	0003B	PUSHL	R6	:	
0000V	CF		05	FB	0003D	CALLS	#5, FILL_FRAME	:	
	6E	0200	8F	3C	00042	MOVZWL	#512, STRING_DESC	:	2696
			5E	DD	00047	PUSHL	SP	:	2701
		04	AE	9F	00049	PUSHAB	STRING_DESC	:	2700
			56	DD	0004C	PUSHL	R6	:	2699
0000V	CF		03	FB	0004E	CALLS	#3, GET_EOJ	:	
			07	DD	00053	PUSHL	#7	:	2706
7E	53		52	C3	00055	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	:	2709
			57	DD	00059	PUSHL	R7	:	2708
50	52		57	C5	0005B	MULL3	R7, TOP_OFFSET, R0	:	2706
			6044	9F	0005F	PUSHAB	(R0)[R4]	:	
		10	AE	9F	00062	PUSHAB	STRING_DESC	:	2705
			56	DD	00065	PUSHL	R6	:	2706
	6B		06	FB	00067	CALLS	#6, INSERT_NAME_BANNER	:	
	55		50	D0	0006A	MOVL	R0, RET_LEN	:	
			05	15	0006D	BLEQ	1\$:	2713
53	52	02	A542	9E	0006F	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	:	2715
	10		05	C3	00074	SUBL3	#5, PAGE_LENGTH, BOTTOM_OFFSET	:	2721
			03	DD	00079	PUSHL	#3	:	2725
			57	DD	0007B	PUSHL	R7	:	
50	53		57	C5	0007D	MULL3	R7, BOTTOM_OFFSET, R0	:	
			6044	9F	00081	PUSHAB	(R0)[R4]	:	
	7E	02A6	C6	9A	00084	MOVZBL	678(R6), -(SP)	:	
			56	DD	00089	PUSHL	R6	:	
0000V	CF		05	FB	0008B	CALLS	#5, FILL_FRAME	:	
	6E	0200	8F	3C	00090	MOVZWL	#512, STRING_DESC	:	2728
			5E	DD	00095	PUSHL	SP	:	2736
		04	AE	9F	00097	PUSHAB	STRING_DESC	:	2735
			56	DD	0009A	PUSHL	R6	:	2734
0000V	CF		03	FB	0009C	CALLS	#3, GET_DIGITAL_LOGO	:	
			01	DD	000A1	PUSHL	#1	:	2740
			57	DD	000A3	PUSHL	R7	:	
	50	01	A3	9E	000A5	MOVAB	1(R3), R0	:	
	50		57	C4	000A9	MULL2	R7, R0	:	
			6044	9F	000AC	PUSHAB	(R0)[R4]	:	
		0C	AE	9F	000AF	PUSHAB	STRING_DESC	:	2739
			56	DD	000B2	PUSHL	R6	:	2740
0000V	CF		05	FB	000B4	CALLS	#5, CENTER_FRAME	:	
	6E	0200	8F	3C	000B9	MOVZWL	#512, STRING_DESC	:	2744
			5E	DD	000BE	PUSHL	SP	:	2749
		04	AE	9F	000C0	PUSHAB	STRING_DESC	:	2748
			56	DD	000C3	PUSHL	R6	:	2747

0000V	CF	03	FB	000C5	CALLS	#3, GET_RULER_COARSE	
		01	DD	000CA	PUSHL	#1	2753
	50	57	DD	000CC	PUSHL	R7	
	50	A3	9E	000CE	MOVAB	4(R3), R0	
		57	C4	000D2	MULL2	R7, R0	
		6044	9F	000D5	PUSHAB	(R0)[R4]	
		0C	AE	9F	000D8	PUSHAB	STRING_DESC
		56	DD	000DB	PUSHL	R6	2752
0000V	CF	05	FB	000DD	CALLS	#5, SCROLL_FRAME	2753
		01	DD	000E2	PUSHL	#1	2757
		57	DD	000E4	PUSHL	R7	
	50	05	A3	9E	000E6	MOVAB	5(R3), R0
	50	57	C4	000EA	MULL2	R7, R0	
		6044	9F	000ED	PUSHAB	(R0)[R4]	
		FF04	CF	9F	000F0	PUSHAB	P.AAE
		56	DD	000F4	PUSHL	R6	2756
0000V	CF	05	FB	000F6	CALLS	#5, SCROLL_FRAME	2757
	6E	0200	8F	3C	000FB	MOVZWL	#512, STRING_DESC
			5E	DD	00100	PUSHL	SP
		04	AE	9F	00102	PUSHAB	STRING_DESC
			7E	D4	00105	CLRL	-(SP)
			56	DD	00107	PUSHL	R6
7E	0000V	CF	04	FB	00109	CALLS	#4, GET_JOB_DESCRIPTION
	53		52	C3	0010E	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
		0090	8F	BB	00112	PUSHR	#*M2R4, R7>
		0C	AE	9F	00116	PUSHAB	STRING_DESC
			56	DD	00119	PUSHL	R6
0000V	CF	05	FB	0011B	CALLS	#5, RETURN_FRAME_LENGTH	2773
	55	50	D0	00120	MOVL	R0, RET_LEN	
		08	15	00123	BLEQ	2\$	2777
50	53	55	C3	00125	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	2779
	53	FF	A0	9E	00129	MOVAB	-1(R0), BOTTOM_OFFSET
			55	DD	0012D	PUSHL	RET_LEN
			57	DD	0012F	PUSHL	R7
50	53		57	C5	00131	MULL3	R7, BOTTOM_OFFSET, R0
		6044	9F	00135	PUSHAB	(R0)[R4]	2785
		0C	AE	9F	00138	PUSHAB	STRING_DESC
			56	DD	0013B	PUSHL	R6
0000V	CF	05	FB	0013D	CALLS	#5, INSERT_FRAME	2784
		07	DD	00142	PUSHL	#7	2785
7E	53		52	C3	00144	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
			57	DD	00148	PUSHL	R7
50	52		57	C5	0014A	MULL3	R7, TOP_OFFSET, R0
		6044	9F	0014E	PUSHAB	(R0)[R4]	2796
		016C	C6	9F	00151	PUSHAB	364(R6)
			56	DD	00155	PUSHL	R6
	6B		06	FB	00157	CALLS	#6, INSERT_NAME_BANNER
	55		50	D0	0015A	MOVL	R0, RET_LEN
			05	15	0015D	BLEQ	3\$
	52	02	A542	9E	0015F	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET
	6E	0200	8F	3C	00164	MOVZWL	#512, STRING_DESC
			5E	DD	00169	PUSHL	SP
		04	AE	9F	0016B	PUSHAB	STRING_DESC
			56	DD	0016E	PUSHL	R6
0000V	CF		03	FB	00170	CALLS	#3, GET_JOB_NAME
			07	DD	00175	PUSHL	#7
7E	53		52	C3	00177	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information into the

B 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 62
(17)

50	52	57	DD	0017B	PUSHL	R7		2823
		57	C5	0017D	MULL3	R7, TOP_OFFSET, R0		2821
		6044	9F	00181	PUSHAB	(R0)[R4]		
	10	AE	9F	00184	PUSHAB	STRING_DESC		2820
		56	DD	00187	PUSHL	R6		2821
	6B	06	FB	00189	CALLS	#6, INSERT_NAME_BANNER		
	55	50	DD	0018C	MOVL	R0, RET_LEN		
		05	15	0018F	BLEQ	4\$		2829
	52	02	A5	42	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET		2831
	58		57	DD	00196	MOVL	R7, RIGHT_OFFSET	2835
			59	D4	00199	CLRL	LEFT_OFFSET	2836
	6E	0200	8F	3C	0019B	MOVZWL	#512, STRING_DESC	2840
			5E	DD	001A0	PUSHL	SP	2845
		04	AE	9F	001A2	PUSHAB	STRING_DESC	2844
			56	DD	001A5	PUSHL	R6	2843
	0000V	CF	03	FB	001A7	CALLS	#3, GET_RECEIPT_BOX	
			08	DD	001AC	PUSHL	#8	2851
7E	58		59	C3	001AE	SUBL3	LEFT_OFFSET, RIGHT_OFFSET, -(SP)	2853
50	52		57	C5	001B2	MULL3	R7, TOP_OFFSET, R0	2851
		6044	9F	001B6	PUSHAB	(R0)[R4]		
		0C	AE	9F	001B9	PUSHAB	STRING_DESC	2850
			56	DD	001BC	PUSHL	R6	2851
	0000V	CF	05	FB	001BE	CALLS	#5, RETURN_FRAME_WIDTH	
	5A		50	DD	001C3	MOVL	R0, RET_WIDE	
7E	53		52	C3	001C6	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2861
		0410	8F	BB	001CA	PUSHR	#M[R4, R10]	2859
		0C	AE	9F	001CE	PUSHAB	STRING_DESC	2858
			56	DD	001D1	PUSHL	R6	2859
	0000V	CF	05	FB	001D3	CALLS	#5, RETURN_FRAME_LENGTH	
	55		50	DD	001D8	MOVL	R0, RET_LEN	
			08	15	001DB	BLEQ	5\$	2863
50	53		55	C3	001DD	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	2865
	53	FF	A0	9E	001E1	MOVAB	-1(R0), BOTTOM_OFFSET	
	58		5A	C2	001E5	SUBL2	RET_WIDE, RIGHT_OFFSET	2868
			55	DD	001E8	PUSHL	RET_LEN	2875
			5A	DD	001EA	PUSHL	RET_WIDE	2874
55	53		57	C5	001EC	MULL3	R7, BOTTOM_OFFSET, R5	2872
50	55		58	C1	001F0	ADDL3	RIGHT_OFFSET, R5, R0	
		6044	9F	001F4	PUSHAB	(R0)[R4]		
		0C	AE	9F	001F7	PUSHAB	STRING_DESC	2871
			56	DD	001FA	PUSHL	R6	2872
	0000V	CF	05	FB	001FC	CALLS	#5, MOVE_FRAME	
	6E	0200	8F	3C	00201	MOVZWL	#512, STRING_DESC	2881
7E	53		52	C3	00206	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2889
7E	58		59	C3	0020A	SUBL3	LEFT_OFFSET, RIGHT_OFFSET, -(SP)	2888
50	55		59	C1	0020E	ADDL3	LEFT_OFFSET, R5, R0	2886
		6044	9F	00212	PUSHAB	(R0)[R4]		
		0C	AE	9F	00215	PUSHAB	STRING_DESC	2885
			56	DD	00218	PUSHL	R6	2886
	0000V	CF	05	FB	0021A	CALLS	#5, INSERT_JOBNUMBER_BANNER	
	6E	0200	8F	3C	0021F	MOVZWL	#512, STRING_DESC	2892
	58		57	DD	00224	MOVL	R7, RIGHT_OFFSET	2893
			59	D4	00227	CLRL	LEFT_OFFSET	2894
			04	00229	RET			2909

; Routine Size: 554 bytes, Routine Base: CODE + 0B58

SEP
V04

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information into the

C 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 63
(17)

SEP
V04

.....
NNNNNNNNNN

: R

```
1980 2910 1 %sbttl 'FILL_FILE_TRAILER - Insert Information into the FILE Page'
1981 2911 1 ++
1982 2912 1 Functional Description:
1983 2913 1 This procedure controls all inserts required for the FILE Page.
1984 2914 1
1985 2915 1 Formal Parameters:
1986 2916 1 SCB - Address of the SCB
1987 2917 1 PAGE_REF - Pointer to the Page (first byte)
1988 2918 1 PAGE_LENGTH - Length of frame
1989 2919 1 PAGE_WIDTH - Width of frame
1990 2920 1
1991 2921 1 Implicit Inputs:
1992 2922 1 none
1993 2923 1
1994 2924 1 Implicit Outputs:
1995 2925 1 none
1996 2926 1
1997 2927 1 Returned Value:
1998 2928 1 none
1999 2929 1
2000 2930 1 Side Effects:
2001 2931 1 none
2002 2932 1 --
2003 2933 1 ROUTINE FILL_FILE_TRAILER (
2004 2934 1 SCB : REF $BBLOCK,
2005 2935 1 PAGE_REF : REF PAGE_ARRAY,
2006 2936 1 PAGE_WIDTH,
2007 2937 1 PAGE_LENGTH
2008 2938 1 ): NOVALUE =
2009 2939 2 BEGIN
2010 2940 2
2011 2941 2 LITERAL K_MAX_BUFFER_SIZE = 512;
2012 2942 2
2013 2943 2 LOCAL
2014 2944 2 FORCE_LEN : VECTOR[1],
2015 2945 2 RET_LEN : VECTOR[1],
2016 2946 2 TOP_OFFSET,
2017 2947 2 BOTTOM_OFFSET,
2018 2948 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
2019 2949 2 STRING_DESC : VECTOR [2]; ! Descriptor to current string
2020 2950 2
2021 2951 2 ! Allocate the buffer for "GET_xxx" Routines
2022 2952 2
2023 2953 2 STRING_DESC[SIZE] = %ALLOCATION(BUFFER); ! allocate for routines
2024 2954 2 STRING_DESC[ADDR] = BUFFER; ! init address
2025 2955 2
2026 2956 2
2027 2957 2 ! Top of page
2028 2958 2
2029 2959 2 TOP_OFFSET = 0;
2030 2960 2 BOTTOM_OFFSET = .PAGE_LENGTH;
2031 2961 2
2032 2962 2 FILL_FRAME (.SCB,
2033 2963 2 .SCB[PSMSB_FILE_BURST_CHAR],
2034 2964 2 PAGE_REF[0, .TOP_OFFSET+2, .PAGE_WIDTH], .PAGE_WIDTH, 3);
2035 2965 2
2036 2966 2 FILL_FRAME (.SCB,
```



```
2037 2967 2 XC' '
2038 2968 PAGE_REF[10,.TOP_OFFSET+2,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);
2039 2969
2040 2970 FILL_FRAME (.SCB,
2041 2971 .SCB[PSM$B_JOB BURST CHAR],
2042 2972 PAGE_REF[13, .TOP_OFFSET+2,.PAGE_WIDTH], .PAGE_WIDTH-26, 3);
2043 2973
2044 2974 ! re-init
2045 2975 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
2046 2976
2047 2977 GET_EOF (.SCB,
2048 2978 STRING_DESC[0], ! Buffer descriptor
2049 2979 STRING_DESC[SIZE]); ! Returned length
2050 2980
2051 2981 RET_LEN[0] = INSERT_NAME_BANNER (
2052 2982 .SCB,
2053 2983 STRING_DESC[SIZE], ! file name desc
2054 2984 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH],
2055 2985 .PAGE_WIDTH, ! ref to frame
2056 2986 .BOTTOM_OFFSET - .TOP_OFFSET, ! max width Bann
2057 2987 ! frame size
2058 2988 7); ! max hght Bann str
2059 2989
2060 2990
2061 2991 ! Adjust for the burst characters too
2062 2992 IF .RET_LEN[0] GTR 0
2063 2993 THEN
2064 2994 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
2065 2995 ! adjust & allow for spacing
2066 2996 ! allow for two spaces...
2067 2997
2068 2998 ! Bottom of page - Bottom_offset already adjusted
2069 2999
2070 3000 BOTTOM_OFFSET = .PAGE_LENGTH - 5; ! offset includes burst offset
2071 3001
2072 3002 FILL_FRAME (.SCB,
2073 3003 .SCB[PSM$B_FILE BURST CHAR],
2074 3004 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
2075 3005
2076 3006 FILL_FRAME (.SCB,
2077 3007 XC' '
2078 3008 PAGE_REF[10,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);
2079 3009
2080 3010 FILL_FRAME (.SCB, ! Offset set... Add the diff
2081 3011 .SCB[PSM$B_JOB BURST CHAR],
2082 3012 PAGE_REF[14,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-28, 3);
2083 3013
2084 3014 ! re-init
2085 3015 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
2086 3016
2087 3017 ! Get the sys$announce note and output to page
2088 3018 GET_DIGITAL_LOGO
2089 3019 (.SCB, ! SCB addr.
2090 3020 STRING_DESC[0], ! Buffer descriptor
2091 3021 STRING_DESC[SIZE]); ! Returned length
2092 3022
2093 3023 ! assume string will not over run the area... fail_safe is truncation
```

```
2094 3024 2 CENTER_FRAME (.SCB,  
2095 3025 STRING_DESC[0],  
2096 3026 PAGE_REF[0,.BOTTOM_OFFSET+1,.PAGE_WIDTH],  
2097 3027 .PAGE_WIDTH, 1);  
2098 3028  
2099 3029  
2100 3030 ! re-init  
2101 3031 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size  
2102 3032  
2103 3033 GET_RULER_COARSE  
2104 3034 (.SCB, ! SCB addr.  
2105 3035 STRING_DESC[0], ! Buffer descriptor  
2106 3036 STRING_DESC[SIZE]); ! Returned length  
2107 3037  
2108 3038  
2109 3039 SCROLL_FRAME (.SCB,  
2110 3040 STRING_DESC[0],  
2111 3041 PAGE_REF[0,.BOTTOM_OFFSET + 4,.PAGE_WIDTH], .PAGE_WIDTH, 1);  
2112 3042  
2113 3043 SCROLL_FRAME (.SCB,  
2114 3044 $DESCRIPTOR ('1234567890'),  
2115 3045 PAGE_REF[0,.BOTTOM_OFFSET + 5,.PAGE_WIDTH], .PAGE_WIDTH, 1);  
2116 3046  
2117 3047 ! Create a sentence describing the current job.  
2118 3048  
2119 3049 ! re-init  
2120 3050 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size  
2121 3051  
2122 3052 GET_JOB_DESCRIPTION  
2123 3053 (.SCB, ! SCB addr.  
2124 3054 0, ! Use past tense  
2125 3055 STRING_DESC[0], ! Buffer descriptor  
2126 3056 STRING_DESC[SIZE]); ! Returned length  
2127 3057  
2128 3058 RET_LEN[0] = RETURN_FRAME_LENGTH  
2129 3059 (.SCB,  
2130 3060 STRING_DESC[0], ! string ref.  
2131 3061 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame  
2132 3062 .PAGE_WIDTH, ! cols to fill  
2133 3063 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill  
2134 3064  
2135 3065 IF .RET_LEN[0] GTR 0  
2136 3066 THEN  
2137 3067 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);  
2138 3068 ! adjust & allow for spacing  
2139 3069 ! before inserting  
2140 3070  
2141 3071 ! insert the string delimited  
2142 3072 INSERT_FRAME (.SCB,  
2143 3073 STRING_DESC[0], ! string ref.  
2144 3074 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], ! ref to frame  
2145 3075 .PAGE_WIDTH, ! cols to fill  
2146 3076 .RET_LEN[0]); ! rows to fill  
2147 3077  
2148 3078 ! Create a sentence describing the current file.  
2149 3079  
2150 3080 ! re-init
```

```
2151 3081 2    STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
2152 3082
2153 3083 2    GET_FILE_DESCRIPTION
2154 3084 2        (.SCB,                                     ! SCB addr.
2155 3085 2        STRING_DESC[0],                             ! Buffer descriptor
2156 3086 2        STRING_DESC[SIZE]);                         ! Returned length
2157 3087
2158 3088 2    RET_LEN[0] = RETURN_FRAME_LENGTH
2159 3089 2        (.SCB,
2160 3090 2        STRING_DESC[0],                             ! string ref.
2161 3091 2        PAGE_REF[0,0,.PAGE_WIDTH],                   ! ref to frame
2162 3092 2        .PAGE_WIDTH,                                   ! cols to fill
2163 3093 2        .BOTTOM_OFFSET - .TOP_OFFSET);                ! rows to fill
2164 3094
2165 3095 2    IF .RET_LEN[0] GTR 0
2166 3096 2    THEN
2167 3097 2        BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
2168 3098 2                                           ! adjust & allow for spacing
2169 3099 2                                           ! before inserting
2170 3100 2    ! insert the string delimited
2171 3101 2    INSERT_FRAME (.SCB,
2172 3102 2        STRING_DESC[0],                             ! string ref.
2173 3103 2        PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH],       ! ref to frame
2174 3104 2        .PAGE_WIDTH,                                   ! cols to fill
2175 3105 2        .RET_LEN[0]);                                ! rows to fill
2176 3106
2177 3107 2    ! User name banner
2178 3108 2    !
2179 3109 2    RET_LEN[0] = INSERT_NAME_BANNER (
2180 3110 2        .SCB,
2181 3111 2        SCB_SIZE (USER NAME),                         ! user name desc
2182 3112 2        PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH],          ! ref to frame
2183 3113 2        .PAGE_WIDTH,                                   ! max width Bann
2184 3114 2        .BOTTOM_OFFSET - .TOP_OFFSET,                 ! frame size
2185 3115 2        7);                                           ! max hght Bann str
2186 3116
2187 3117 2    IF .RET_LEN[0] GTR 0
2188 3118 2    THEN
2189 3119 2        TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
2190 3120 2                                           ! adjust & allow for spacing
2191 3121 2
2192 3122 2    ! Get and insert the filename banner ... force the banner to be small (always)
2193 3123 2    !
2194 3124 2    ! re-init
2195 3125 2    STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
2196 3126 2
2197 3127 2    FORCE_LEN = 7;
2198 3128 2    IF .BOTTOM_OFFSET - .TOP_OFFSET LSS .FORCE_LEN
2199 3129 2    THEN
2200 3130 2        FORCE_LEN = .BOTTOM_OFFSET - .TOP_OFFSET;
2201 3131 2
2202 3132 2    RET_LEN[0] = INSERT_FILENAME_BANNER
2203 3133 2        (.SCB,
2204 3134 2        STRING_DESC[0],                             ! file name size
2205 3135 2
2206 3136 2
2207 3137 2
```

```
2208 3138 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH],
2209 3139 ! ref to frame
2210 3140 ! max width Bann
2211 3141 ! max hght Bann str
2212 3142
2213 3143 IF .RET_LEN[0] GTR 0
2214 3144 THEN
2215 3145 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
2216 3146 ! adjust & allow for spacing
2217 3147
2218 3148 ! Create a phrase which includes all the appropriate qualifiers
2219 3149 describing the current print and insert from the bottom without spacing.
2220 3150
2221 3151 re-init
2222 3152 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
2223 3153
2224 3154 GET_QUALIFIERS
2225 3155 (.SCB, ! SCB addr.
2226 3156 STRING_DESC[0], ! Buffer descriptor
2227 3157 STRING_DESC[SIZE]); ! Returned length
2228 3158
2229 3159 RET_LEN[0] = RETURN_FRAME_LENGTH
2230 3160 (.SCB,
2231 3161 STRING_DESC[0], ! string ref.
2232 3162 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
2233 3163 .PAGE_WIDTH-12, ! less twelve chars.
2234 3164 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
2235 3165
2236 3166 IF .RET_LEN[0] GTR 0
2237 3167 THEN
2238 3168 BEGIN
2239 3169 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
2240 3170 ! adjust & allow for spacing
2241 3171 ! before inserting
2242 3172
2243 3173 ! move the string undelimited
2244 3174 MOVE_FRAME (.SCB,
2245 3175 $DESCRIPTOR('Qualifiers: '),
2246 3176 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH],
2247 3177 ! ref to frame
2248 3178 .PAGE_WIDTH, ! cols to fill
2249 3179 .RET_LEN[0]); ! rows to fill
2250 3180 END;
2251 3181
2252 3182 ! insert the string delimited
2253 3183 INSERT_FRAME (.SCB,
2254 3184 STRING_DESC[0], ! string ref.
2255 3185 PAGE_REF[12,.BOTTOM_OFFSET,.PAGE_WIDTH],
2256 3186 ! ref to frame
2257 3187 .PAGE_WIDTH-12, ! cols to fill
2258 3188 .RET_LEN[0]); ! rows to fill
2259 3189
2260 3190 1 END;
```


20 3A 73 72 65 69 66 69 6C 61 75 51 00D8C P.AAG: .LONG 10
00000000' 00D90 .ADDRESS P.AAH
00D94 P.AAJ: .ASCII \Qualifiers: \
0000000C 00DA0 P.AAI: .LONG 12
00000000' 00DA4 .ADDRESS P.AAJ

OFFC 00000 FILL_FILE_TRAILER:

5B	0000V	CF	9E	00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	2933
5A	0000V	CF	9E	00007	MOVAB	INSERT_FRAME, R11	
59	0000V	CF	9E	0000C	MOVAB	RETURN_FRAME_LENGTH, R10	
5E	FDFC	CE	9E	00011	MOVAB	FILL_FRAME, R9	
7E	0200	8F	3C	00016	MOVZWL	-516(SP), SP	2953
04	AE	08	AE	0001B	MOVAB	#512, STRING_DESC	2954
		52	D4	00020	MOVAB	BUFFER, STRING_DESC+4	2959
53	10	AC	D0	00022	CLRL	TOP_OFFSET	2960
		03	DD	00026	MOVL	PAGE_LENGTH, BOTTOM_OFFSET	2964
56	0C	AC	D0	00028	PUSHL	#3	
		56	DD	0002C	MOVL	PAGE_WIDTH, R6	
54	08	AC	D0	0002E	PUSHL	R6	
55	02	A2	9E	00032	MOVL	PAGE_REF, R4	
55		56	C4	00036	MOVAB	2(R2), R5	
		6544	9F	00039	MULL2	R6, R5	
58	04	AC	D0	0003C	PUSHAB	(R5)[R4]	
7E	02A4	C8	9A	00040	MOVL	SCB, R8	2963
		58	DD	00045	MOVZBL	676(R8), -(SP)	2964
69		05	FB	00047	PUSHL	R8	
		03	DD	0004A	CALLS	#5, FILL_FRAME	
	EC	A6	9F	0004C	PUSHL	#3	2968
	0A	A544	9F	0004F	PUSHAB	-20(R6)	
		20	DD	00053	PUSHAB	10(R5)[R4]	
		58	DD	00055	PUSHL	#32	
69		05	FB	00057	PUSHL	R8	
		03	DD	0005A	CALLS	#5, FILL_FRAME	
	E6	A6	9F	0005C	PUSHL	#3	2972
	0D	A544	9F	0005F	PUSHAB	-26(R6)	
7E	02A6	C8	9A	00063	PUSHAB	13(R5)[R4]	
		58	DD	00068	MOVZBL	678(R8), -(SP)	
69		05	FB	0006A	PUSHL	R8	
6E	0200	8F	3C	0006D	CALLS	#5, FILL_FRAME	
		5E	DD	00072	MOVZWL	#512, STRING_DESC	2975
	04	AE	9F	00074	PUSHL	SP	2979
		58	DD	00077	PUSHAB	STRING_DESC	2978
0000V	CF	03	FB	00079	PUSHL	R8	2977
		07	DD	0007E	CALLS	#3, GET_EOF	
7E	53	52	C3	00080	PUSHL	#7	2984
		56	DD	00084	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2987
55	52	56	C5	00086	PUSHL	R6	2986
		6544	9F	0008A	MULL3	R6, TOP_OFFSET, R5	2984
	10	AE	9F	0008D	PUSHAB	(R5)[R4]	
		58	DD	00090	PUSHAB	STRING_DESC	2983
0000V	CF	06	FB	00092	PUSHL	R8	2984
57		50	D0	00097	CALLS	#6, INSERT_NAME_BANNER	
		05	15	0009A	MOVL	R0, RET_LEN	
52	02	A742	9E	0009C	BLEQ	18	2992
					MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	2994

```
Print Symbiont -- separation routines
FILL_FILE_TRAILER - Insert Information into the
```

J 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 70
(18)

SEP
V04

Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419
---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

7E	0000V	CF	53	0050	7E D4 00153	CLRL	-(SP)	3052
				0C	58 DD 00155	PUSHL	R8	3052
		6A	57		04 FB 0C157	CALLS	#4, GET_JOB_DESCRIPTION	3062
					52 C3 0015C	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	3060
					8F BB 00160	PUSHR	#MZR4, R6>	3059
					AE 9F 00164	PUSHAB	STRING_DESC	3060
					58 DD 00167	PUSHL	R8	3064
					05 FB 00169	CALLS	#5, RETURN_FRAME_LENGTH	3066
					50 D0 0016C	MOVL	R0, RET_LEN	3064
					08 15 0016F	BLEQ	2\$	3066
50		53			57 C3 00171	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	3074
		53		FF	A0 9E 00175	MOVAB	-1(R0), BOTTOM_OFFSET	3072
		7E			56 7D 00179	MOVQ	R6, -(SP)	3072
50		53			56 C5 0017C	MULL3	R6, BOTTOM_OFFSET, R0	3071
				6044	9F 00180	PUSHAB	(R0)[R4]	3072
				0C	AE 9F 00183	PUSHAB	STRING_DESC	3081
					58 DD 00186	PUSHL	R8	3086
		6B			05 FB 00188	CALLS	#5, INSERT_FRAME	3085
		6E		0200	8F 3C 0018B	MOVZWL	#512, STRING_DESC	3084
				04	5E DD 00190	PUSHL	SP	3093
					AE 9F 00192	PUSHAB	STRING_DESC	3091
					58 DD 00195	PUSHL	R8	3090
7E	0000V	CF	53	0050	03 FB 00197	CALLS	#3, GET_FILE_DESCRIPTION	3091
				0C	52 C3 0019C	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	3091
					8F BB 001A0	PUSHR	#MZR4, R6>	3095
					AE 9F 001A4	PUSHAB	STRING_DESC	3097
					58 DD 001A7	PUSHL	R8	3095
		6A			05 FB 001A9	CALLS	#5, RETURN_FRAME_LENGTH	3097
		57			50 D0 001AC	MOVL	R0, RET_LEN	3105
					08 15 001AF	BLEQ	3\$	3103
50		53			57 C3 001B1	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	3102
		53		FF	A0 9E 001B5	MOVAB	-1(R0), BOTTOM_OFFSET	3103
		7E			56 7D 001B9	MOVQ	R6, -(SP)	3113
50		53			56 C5 001BC	MULL3	R6, BOTTOM_OFFSET, R0	3116
				6044	9F 001C0	PUSHAB	(R0)[R4]	3115
				0C	AE 9F 001C3	PUSHAB	STRING_DESC	3112
					58 DD 001C6	PUSHL	R8	3113
		6B			05 FB 001C8	CALLS	#5, INSERT_FRAME	3120
					07 DD 001CB	PUSHL	#7	3122
7E		53			52 C3 001CD	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	3128
					56 DD 001D1	PUSHL	R6	3130
55		52			56 C5 001D3	MULL3	R6, TOP_OFFSET, R5	3131
				6544	9F 001D7	PUSHAB	(R5)[R4]	3112
				016C	C8 9F 001DA	PUSHAB	364(R8)	3113
					58 DD 001DE	PUSHL	R8	3120
	0000V	CF	57		06 FB 001E0	CALLS	#6, INSERT_NAME_BANNER	3122
					50 D0 001E5	MOVL	R0, RET_LEN	3128
					05 15 001E8	BLEQ	4\$	3130
		52		02	A742 9E 001EA	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	3131
		6E		0200	8F 3C 001EF	MOVZWL	#512, STRING_DESC	3133
		51			07 D0 001F4	MOVL	#7, FORCE_LEN	3133
50		53			52 C3 001F7	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, R0	3133
		51			50 D1 001FB	CMPL	R0, FORCE_LEN	3141
					03 18 001FE	BGEQ	5\$	3140
		51			50 D0 00200	MOVL	R0, FORCE_LEN	3140
					51 DD 00203	PUSHL	FORCE_LEN	3140
					56 DD 00205	PUSHL	R6	3140

```
Print Symbiont -- separation routines
FILL_FILE_TRAILER - Insert Information into the
```

L²
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 72
(18)SEP
V04

55		52		56 C5 00207	MULL3 R6, TOP_OFFSET, R5	: 3138
			OC	6544 9F 0020B	PUSHAB (R5)[R4]	:
				AE 9F 0020E	PUSHAB STRING_DESC	: 3137
				58 DD 00211	PUSHL R8	: 3138
	0000V	CF		05 FB 00213	CALLS #5, INSERT_FILENAME_BANNER	:
		57		50 D0 00218	MOVL R0, RET_LEN	:
				05 15 0021B	BLEQ 6\$: 3143
		52	02 A742	9E 0021D	MOVAB 2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	: 3145
		6E	0200	8F 3C 00222	MOVZWL #512, -STRING_DESC	: 3152
				5E DD 00227	PUSHL SP	: 3157
			04	AE 9F 00229	PUSHAB STRING_DESC	: 3156
				58 DD 0022C	PUSHL R8	: 3155
	0000V	CF		03 FB 0022E	CALLS #3, GET_QUALIFIERS	:
7E		53		52 C3 00233	SUBL3 TOP_OFFSET, BOTTOM_OFFSET, -(SP)	: 3164
			F4	A6 9F 00237	PUSHAB -12(R6)	: 3163
				54 DD 0023A	PUSHL R4	: 3162
			OC	AE 9F 0023C	PUSHAB STRING_DESC	: 3161
				58 DD 0023F	PUSHL R8	: 3162
		6A		05 FB 00241	CALLS #5, RETURN_FRAME_LENGTH	:
		57		50 D0 00244	MOVL R0, RET_LEN	:
				1D 15 00247	BLEQ 7\$: 3166
50		53		57 C3 00249	SUBL3 RET_LEN, BOTTOM_OFFSET, R0	: 3169
		53	FF	A0 9E 0024D	MOVAB -1(R0), BOTTOM_OFFSET	:
		7E		56 7D 00251	MOVQ R6, -(SP)	: 3178
50		53		56 C5 00254	MULL3 R6, BOTTOM_OFFSET, R0	: 3176
			6044	9F 00258	PUSHAB (R0)[R4]	:
			FD99	CF 9F 0025B	PUSHAB P.AAI	: 3175
				58 DD 0025F	PUSHL R8	: 3176
	0000V	CF		05 FB 00261	CALLS #5, MOVE_FRAME	:
				57 DD 00266	PUSHL RET_LEN	: 3188
			F4	A6 9F 00268	PUSHAB -12(R6)	: 3187
		53		56 C4 0026B	MULL2 R6, R3	: 3185
			OC A344	9F 0026E	PUSHAB 12(R3)[R4]	:
			OC	AE 9F 00272	PUSHAB STRING_DESC	: 3184
				58 DD 00275	PUSHL R8	: 3185
		6B		05 FB 00277	CALLS #5, INSERT_FRAME	:
				04 0027A	RET	: 3190

; Routine Size: 635 bytes, Routine Base: CODE + 0DA8


```

2262 3191 1 %sbtll 'RETURN_FRAME_LENGTH - Returns the Frame Length for String Insertion'
2263 3192 1 ++
2264 3193 1 Functional Description:
2265 3194 1 Returns the frame length needed to insert the string into the page.
2266 3195 1 This routine checks the top_of_frame/bottom_of_frame offsets and
2267 3196 1 decides if the string will fit=
2268 3197 1 1) yes - return number of frame rows required for string to fit.
2269 3198 1 2) no - return zero
2270 3199 1
2271 3200 1
2272 3201 1 Formal Parameters:
2273 3202 1 SCB - Address of the SCB
2274 3203 1 STR_DESC - Descriptor of String to Insert
2275 3204 1 FRAME_PTR - Address of first byte of frame
2276 3205 1 FRAME_LENGTH - Length of frame
2277 3206 1 FRAME_WIDTH - Width of frame
2278 3207 1
2279 3208 1
2280 3209 1 Implicit Inputs:
2281 3210 1 none
2282 3211 1
2283 3212 1 Implicit Outputs:
2284 3213 1 none
2285 3214 1
2286 3215 1 Returned Value:
2287 3216 1 none
2288 3217 1
2289 3218 1 Side Effects:
2290 3219 1 none
2291 3220 1 --
2292 3221 1 ROUTINE RETURN_FRAME_LENGTH (
2293 3222 1 SCB : REF $BBLOCK,
2294 3223 1 STR_DESC : REF VECTOR[2],
2295 3224 1 FRAME_PTR : REF PAGE_ARRAY,
2296 3225 1 FRAME_WIDTH , Number of Columns
2297 3226 1 FRAME_LENGTH ) = Number of Rows
2298 3227 1
2299 3228 2 BEGIN
2300 3229 2
2301 3230 2 LOCAL MAX_CHARS;
2302 3231 2
2303 3232 2 ! don't even try if there is no frame left
2304 3233 2 IF (.FRAME_LENGTH LEQ 0) OR
2305 3234 2 (.FRAME_WIDTH LEQ 0) OR
2306 3235 2 (.STR_DESC[SIZE] EQL 0)
2307 3236 2 THEN
2308 3237 2 RETURN 0;
2309 3238 2
2310 3239 2 MAX_CHARS = .FRAME_WIDTH * .FRAME_LENGTH;
2311 3240 2
2312 3241 2 ! The boundary condition of string size of some multiple of frame width
2313 3242 2 can occur - add one less than the frame width to overcome this condition
2314 3243 2
2315 3244 2 IF .STR_DESC[SIZE] LEQ .MAX_CHARS
2316 3245 2 THEN
2317 3246 2 RETURN ((.STR_DESC[SIZE]+(.FRAME_WIDTH-1)) / .FRAME_WIDTH);
2318 3247 2

```

SEPARATE
V04-001

Print Symbiont -- separation routines
RETURN_FRAME_LENGTH - Returns the Frame Length

N 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 74
(19)

: 2319
: 2320
: 2321
: 2322

3248 2 ! otherwise it just won't fit
3249 2 RETURN 0;
3250 2
3251 1 END;

0000 00000 RETURN_FRAME_LENGTH:

			14	AC	D5	00002	.WORD	Save nothing	: 3221
			23	15	00005	TSTL	FRAME_LENGTH		: 3233
			10	AC	D5	00007	BLEQ	1\$	
			1E	15	0000A	TSTL	FRAME_WIDTH		: 3234
			08	BC	D5	0000C	BLEQ	1\$	
			19	13	0000F	TSTL	@STR_DESC		: 3235
50	10	AC	14	AC	C5	00011	BEQL	1\$	
		50	08	BC	D1	00017	MULL3	FRAME_LENGTH, FRAME_WIDTH, MAX_CHARS	: 3239
			0D	14	0001B	CMPL	@STR_DESC, MAX_CHARS		: 3244
50	08	BC	10	AC	C1	0001D	BGTR	1\$	
			50	D7	00023	ADDL3	FRAME_WIDTH, @STR_DESC, R0		: 3246
		50	10	AC	C6	00025	DECL	R0	
					04	00029	DIVL2	FRAME_WIDTH, R0	
			50	D4	0002A	1\$:	RET		: 3251
					04	0002C	CLRL	R0	
							RET		

; Routine Size: 45 bytes, Routine Base: CODE + 1023

```
2324 3252 1 %sbttl 'RETURN_FRAME_WIDTH - Returns the Frame Length for String Insertion'
2325 3253 1 ++
2326 3254 1 Functional Description:
2327 3255 1 Returns the frame width needed to insert the required lengths into
2328 3256 1 the page. This routine returns only a prescribed value and
2329 3257 1 decides if the string will fit -
2330 3258 1 Return value - frame width
2331 3259 1 Return zero - only if no length or width of frame.
2332 3260 1 Assumes the FRAME_LENGTH is constant. (How many lengths are needed
2333 3261 1 to fit this string)
2334 3262 1
2335 3263 1 Formal Parameters:
2336 3264 1 SCB - Address of the SCB
2337 3265 1 STR_DESC - Descriptor of String to Insert
2338 3266 1 FRAME_PTR - Address of first byte of frame
2339 3267 1 FRAME_LENGTH - Length of frame
2340 3268 1 FRAME_WIDTH - Width of frame
2341 3269 1
2342 3270 1
2343 3271 1 Implicit Inputs:
2344 3272 1 none
2345 3273 1
2346 3274 1 Implicit Outputs:
2347 3275 1 none
2348 3276 1
2349 3277 1 Returned Value:
2350 3278 1 none
2351 3279 1
2352 3280 1 Side Effects:
2353 3281 1 none
2354 3282 1 --
2355 3283 1 ROUTINE RETURN_FRAME_WIDTH (
2356 3284 1 SCB : REF $BBLOCK,
2357 3285 1 STR_DESC : REF VECTOR[2],
2358 3286 1 FRAME_PTR : REF PAGE_ARRAY,
2359 3287 1 FRAME_WIDTH , ! Number of Columns
2360 3288 1 FRAME_LENGTH ! Number of Rows
2361 3289 1 ) =
2362 3290 2 BEGIN
2363 3291 2
2364 3292 2 LOCAL
2365 3293 2 MAX_CHARS,
2366 3294 2 TEMP_WIDE;
2367 3295 2
2368 3296 2 ! don't even try if there is no frame left
2369 3297 2 IF (.FRAME_LENGTH LEQ 0) OR
2370 3298 2 (.FRAME_WIDTH LEQ 0) OR
2371 3299 2 (.STR_DESC[SIZE] EQL 0)
2372 3300 2 THEN
2373 3301 2 RETURN 0;
2374 3302 2
2375 3303 2 MAX_CHARS = .FRAME_WIDTH * .FRAME_LENGTH;
2376 3304 2 IF .MAX_CHARS LSS .STR_DESC[SIZE]
2377 3305 2 THEN
2378 3306 2 RETURN 0; ! string wont fit
2379 3307 2
2380 3308 2 ! The boundary condition of string size of some multiple of frame width
```

; Routine Size: 43 bytes, Routine Base: CODE + 1050


```
2394 3321 1 %sbttl 'GET_REVISION_DATE - Get the revision date of current file'
2395 3322 1
2396 3323 1 *+
2397 3324 1 Functional Description:
2398 3325 1 This routine creates a phrase with DD-MMM-YYYY HH:MM describing
2399 3326 1 the revision date of the current file. Returns zero if file
2400 3327 1 not open.
2401 3328 1
2402 3329 1 Formal Parameters:
2403 3330 1 SCB - Address of the SCB
2404 3331 1 STR_DESC - Desc of String to Return
2405 3332 1 RET_LEN - Return length of Desc.
2406 3333 1
2407 3334 1 Implicit Inputs:
2408 3335 1 none
2409 3336 1
2410 3337 1 Implicit Outputs:
2411 3338 1 none
2412 3339 1
2413 3340 1 Returned Value:
2414 3341 1 none
2415 3342 1
2416 3343 1 Side Effects:
2417 3344 1 none
2418 3345 1 --
2419 3346 1 ROUTINE GET_REVISION_DATE (
2420 3347 1 SCB : REF $BBLOCK, ! SCB
2421 3348 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
2422 3349 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
2423 3350 1 ) : NOVALUE =
2424 3351 2 BEGIN
2425 3352 2 BIND
2426 3353 2 XABDAT = .SCB[PSM$A_XABDAT]: $BBLOCK, ! - RMS date block
2427 3354 2 FORMAT = $DESCRIPTOR ( ! - revision date
2428 3355 2 '!17XD' ),
2429 3356 2
2430 3357 2 NAM = .SCB[PSM$A_NAM]: REF $BBLOCK;
2431 3358 2
2432 3359 2 LOCAL
2433 3360 2 CURRENT_LEN : INITIAL (0);
2434 3361 2
2435 3362 2 IF FILE_OPEN(.SCB)
2436 3363 2 THEN
2437 3364 2 $FAO (
2438 3365 2 FORMAT,
2439 3366 2 CURRENT_LEN,
2440 3367 2 STR_DESC[0],
2441 3368 2 XABDAT[XAB$Q_RDT],
2442 3369 2 );
2443 3370 2 RET_LEN[0] = .CURRENT_LEN;
2444 3371 2
2445 3372 1 END;
```

```
44 25 37 31 21 0107B P.AAL: .ASCII \!17XD\
00000005 01080 P.AAK: .LONG 5
```

SEPARATE
V04-001

Print Symbiont -- separation routines

GET_REVISION_DATE - Get the revision date of cu

E 3

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742

[PRTSMB.SRC]SEPARATE.B32;2

Page 78
(21)

00000000' 01084

.ADDRESS P.AAL

FORMAT=

P.AAK

0004 00000 GET_REVISION_DATE:

50	04	AC	D0	00002	.WORD	Save R2	: 3345
52	0254	C0	D0	00006	MOVL	SCB, R0	: 3352
		7E	D4	0000B	MOVL	596(R0), R2	: 3356
		50	DD	0000D	CLRL	CURRENT_LEN	: 3361
0000V	CF	01	FB	0000F	PUSHL	R0	: 3361
	13	50	E9	00014	CALLS	#1, FILE_OPEN	: 3368
		0C	A2	9F	BLBC	R0, 1\$: 3368
		08	AC	DD	PUSHAB	12(R2)	: 3368
		08	AE	9F	PUSHL	STR_DESC	: 3368
		D5	AF	9F	PUSHAB	CURRENT_LEN	: 3368
00000000G	00	04	FB	00023	PUSHAB	FORMAT	: 3368
	0C	6E	B0	0002A	CALLS	#4, SYSSFA0	: 3370
	BC	04	00	0002E	MOVW	CURRENT_LEN, @RET_LEN	: 3372
					RET		: 3372

; Routine Size: 47 bytes, Routine Base: CODE + 1088

```
2447 3373 1 %sbttl 'GET_SYSTEM_ANNOUNCEMENT- Create a Sentence Describing the Current Job'
2448 3374 1 ++
2449 3375 1 Functional Description:
2450 3376 1 This routine get the system annoucement. All allocation of buffers
2451 3377 1 handled by caller
2452 3378 1
2453 3379 1 Formal Parameters:
2454 3380 1 SCB - Address of the SCB
2455 3381 1 STR_DESC - Desc of String to Return
2456 3382 1 RET_LEN - Return length of Desc.
2457 3383 1
2458 3384 1 Implicit Inputs:
2459 3385 1 none
2460 3386 1
2461 3387 1 Implicit Outputs:
2462 3388 1 none
2463 3389 1
2464 3390 1 Returned Value:
2465 3391 1 none
2466 3392 1
2467 3393 1 Side Effects:
2468 3394 1 none
2469 3395 1 --
2470 3396 1 ROUTINE GET_SYSTEM_ANNOUNCEMENT (
2471 3397 1 SCB : REF $BBLOCK, ! SCB
2472 3398 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
2473 3399 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
2474 3400 1 ) : NOVALUE =
2475 3401 2 BEGIN
2476 3402 2 BIND
2477 P 3403 2 DEFINED_ANNOUNCE = $DESCRIPTOR (
2478 3404 2 '!AF' ), ! - Defined announcement
2479 3405 2
2480 P 3406 2 ANNOUNCE = $DESCRIPTOR (
2481 3407 2 'PSM$ANNOUNCE' ); ! - system announcement
2482 3408 2
2483 3409 2 LOCAL
2484 3410 2 FAO_DESC : VECTOR[2],
2485 3411 2 BUFFER : VECTOR[256,byte],
2486 3412 2 TEMP_LEN : INITIAL (0),
2487 3413 2 STATUS :
2488 3414 2
2489 3415 2 FAO_DESC[SIZE] = %ALLOCATION(BUFFER);
2490 3416 2 FAO_DESC[ADDR] = BUFFER;
2491 3417 2
2492 3418 2 !STATUS = $TRNLNM(attr = LNM$M_CASE BLIND,
2493 3419 2 tabnam=%ASCII-'LNM$SYSTEM_TABLE',
2494 3420 2 lognam= ANNOUNCE,
2495 3421 2 rsllen= FAO_DESC[SIZE],
2496 3422 2 rsbuf= FAO_DESC[ADDR]);
2497 3423 2
2498 P 3424 2 STATUS = $TRNLOG(lognam= ANNOUNCE,
2499 P 3425 2 rsbuf= FAO_DESC,
2500 3426 2 rsllen= TEMP_LEN);
2501 3427 2
2502 3428 2 IF .STATUS
2503 3429 2 THEN ! Success - Normal, Buffer_overflow
```

```
2504 3430 2 |
2505 3431 2 | check for command file pointer 'a' sign or no-translation code
2506 3432 2 |
2507 3433 2 BEGIN
2508 3434 2 IF (.STATUS EQL SSS_NOTRAN)
2509 3435 2 THEN
2510 3436 2 TEMP_LEN = 0;
2511 3437 2 END
2512 3438 2 ELSE | Bad status - Badparam, Badlength,
2513 3439 2 TEMP_LEN = 0; | Badtable, Notfound(badname), Baddepth
2514 3440 2
2515 3441 2 IF .TEMP_LEN EQL 0 | Get the default sys version #
2516 3442 2 THEN
2517 3443 2 ! Put in the Digital Logo
2518 3444 2 GET_DIGITAL_LOGO(.SCB,STR_DESC[0],RSL_LEN[0])
2519 3445 2 ELSE
2520 3446 2 $FAO (
2521 3447 2 DEFINED_ANNOUNCE,
2522 3448 2 RSL_LEN[0],
2523 3449 2 STR_DESC[0],
2524 3450 2 .TEMP_LEN,
2525 3451 2 .FAO_DESC[ADDR]);
2526 3452 2
2527 3453 1 END;
```

```
46 41 21 010B7 P.AAN: .ASCII \!AF\
010BA .BLKB 2
00000003 010BC P.AAM: .LONG 3
00000000 010C0 .ADDRESS P.AAN
45 43 4E 55 4F 4E 4E 41 24 4D 53 50 010C4 P.AAP: .ASCII \PSM$ANNOUNCE\
0000000C 010D0 P.AAO: .LONG 12
00000000 010D4 .ADDRESS P.AAP
```

```
DEFINED_ANNOUNCE= P.AAM
ANNOUNCE= P.AAO
.EXTRN SYS$TRNLOG
```

```
0000 00000 GET_SYSTEM_ANNOUNCEMENT:
SE FEF8 CE 9E 00002 .WORD Save nothing 3396
7E D4 00007 MOVAB -264(SP), SP
F8 AD 0100 8F 3C 00009 CLRL TEMP_LEN 3401
FC AD 04 AE 9E 0000F MOVZWL #256, FAO_DESC 3415
7E 7C 00014 MOVAB BUFFER, FAO_DESC+4 3416
7E D4 00016 CLRL -(SP) 3426
F8 AD 9F 00018 CLRL -(SP)
10 AE 9F 0001B PUSHAB FAO_DESC
D7 AF 9F 0001E PUSHAB TEMP_LEN
00000000G 00 06 FB 00021 PUSHAB ANNOUNCE
09 50 E9 00028 CALLS #6, SYS$TRNLOG 3428
00000629 8F 50 D1 0002B BLBC STATUS, 1$ 3434
02 12 00032 CMPL STATUS, #1577
6E D4 00034 1$: CLRL TEMP_LEN 3439
6E D5 00036 2$: TSTL TEMP_LEN 3441
OD 12 00038 BNEQ 3$
```


SEPARATE
V04-001

Print Symbiont -- separation routines
GET_SYSTEM_ANNOUNCEMENT- Create a Sentence Desc

H 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 81
(22)

7E 08 AC 7D 0003A
04 AC DD 0003E
0000V CF 03 FB 00041
04 00046
FC AD DD 00047 3\$:
04 AE DD 0004A
08 AC DD 0004D
0C AC DD 00050
8E AF 9F 00053
00000000G 00 05 FB 00056
04 0005D

MOVQ STR_DESC, -(SP)
PUSHL SCB
CALLS #3, GET_DIGITAL_LOGO
RET
PUSHL FAO_DESC+4
PUSHL TEMP_LEN
PUSHL STR_DESC
PUSHL RSL_LEN
PUSHAB DEFINED_ANNOUNCE
CALLS #5, SYS\$FAO
RET

3444
3451
3453

; Routine Size: 94 bytes, Routine Base: CODE + 10D8

```
2529 3454 1 %sbtll 'GET_VMS_LOGO - Create a Phrase of VMS Logo'
2530 3455 1 ++
2531 3456 1 Functional Description:
2532 3457 1 VAX/VMS Version Vx.x
2533 3458 1
2534 3459 1 Formal Parameters:
2535 3460 1 SCB - Address of the SCB
2536 3461 1 STR_DESC - Desc of String to Return
2537 3462 1 RET_LEN - Return length of Desc.
2538 3463 1
2539 3464 1 Implicit Inputs:
2540 3465 1 none
2541 3466 1
2542 3467 1 Implicit Outputs:
2543 3468 1 none
2544 3469 1
2545 3470 1 Returned Value:
2546 3471 1 none
2547 3472 1
2548 3473 1 Side Effects:
2549 3474 1 none
2550 3475 1 --
2551 3476 1 ROUTINE GET_VMS_LOGO (
2552 3477 1 SCB : REF $BLOCK, ! SCB
2553 3478 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
2554 3479 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
2555 3480 1 ) : NOVALUE =
2556 3481 2 BEGIN
2557 3482 2 BIND
2558 3483 2 TRAILING = 1,
2559 3484 2
2560 3485 2 DEFAULT = $DESCRIPTOR (
2561 3486 2 'VAX/VMS'
2562 3487 2 'VAX/VMS'
2563 3488 2 'VAX/VMS'
2564 3489 2 'VAX/VMS'
2565 3490 2 'VAX/VMS'
2566 3491 2 'VAX/VMS'
2567 3492 2 'VAX/VMS'
2568 3493 2 'VAX/VMS'
2569 3494 2 'VAX/VMS'
2570 3495 2 'VAX/VMS'
2571 3496 2 'VAX/VMS'
2572 3497 2 'VAX/VMS'
2573 3498 2 'VAX/VMS'
2574 3499 2 'VAX/VMS'
2575 3500 2 'VAX/VMS'
2576 3501 2 'VAX/VMS'
2577 3502 2 'VAX/VMS'
2578 3503 2 'VAX/VMS'
2579 3504 2 'VAX/VMS'
2580 3505 2 'VAX/VMS'
2581 3506 2 'VAX/VMS'
2582 3507 2 'VAX/VMS'
2583 3508 2 'VAX/VMS'
2584 3509 2 'VAX/VMS'
2585 3510 2 'VAX/VMS' );
```

```
2586 3511 LOCAL
2587 3512 STR_PTR
2588 3513 STR_LEN
2589 3514
2590 3515
2591 3516 IF .SCB[PSM$L_PAGE_WIDTH] LSS 20
2592 3517 THEN ! no room for burst bar
2593 3518 BEGIN
2594 3519 RSL_LEN[0] = 0;
2595 3520 RETURN;
2596 3521 END;
2597 3522
2598 3523 $FAO (
2599 3524 DEFAULT
2600 3525 RSL_LEN[0],
2601 3526 STR_DESC[0]);
2602 3527
2603 3528 RSL_LEN[0] = .SCB[PSM$L_PAGE_WIDTH] - 20; ! set the page length
2604 3529 ! largest less than 180
2605 3530 STR_PTR = CH$PTR(.STR_DESC[ADDR]+.RSL_LEN[0]);
2606 3531
2607 3532 WHILE CH$NEQ( 1, .STR_PTR, 1, CH$PTR(UPLIT(' ')))
2608 3533 DO ! trim off chars until blanks
2609 3534 BEGIN
2610 3535 RSL_LEN[0] = .RSL_LEN[0] - 1;
2611 3536 STR_PTR = CH$PTR(.STR_DESC[ADDR]+.RSL_LEN[0]);
2612 3537 END;
2613 3538
2614 3539 END;
```

20	20	53	4D	56	2F	58	41	56	01136	P.AAR:	.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	0113F		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01148		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01151		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	0115A		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01163		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	0116C		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01175		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	0117E		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01187		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01190		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01199		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011A2		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011AB		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011B4		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011BD		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011C6		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011CF		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011D8		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011E1		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011EA		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011F3		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	011FC		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	01205		.ASCII	\VAX/VMS	/
20	20	53	4D	56	2F	58	41	56	0120E		.ASCII	\VAX/VMS	/

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_VMS_LOGO - Create a Phrase of VMS logo

K 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 84
(23)

00 00 00 20 01215 .BLKB 3
000000DF 01218 P.AAQ: .LONG 223
00000000 0121C .ADDRESS P.AAR
01220 P.AAS: .ASCII \ \<0><0><0>

TRAILING= 1
DEFAULT= P.AAQ

001C 00000 GET_VMS_LOGO:						
53	04	AC	D0 00002	.WORD	Save R2,R3,R4	3476
14	0200	C3	D1 00006	MOVL	SCB, R3	3516
		04	18 0000B	CMPL	512(R3), #20	
	0C	BC	B4 0000D	BGEQ	1\$	
			04 00010	CLRW	@RSL_LEN	3519
54	08	AC	D0 00011 1\$:	RET		3518
		54	DD 00015	MOVL	STR_DESC, R4	3526
52	0C	AC	D0 00017	PUSHL	R4	
		52	DD 0001B	MOVL	RSL_LEN, R2	
	D4	AF	9F 0001D	PUSHL	R2	
62 00000000G 00		03	FB 00020	PUSHAB	DEFAULT	
0200 C3		14	A3 00027	CALLS	#3, SYSSFA0	
50		62	3C 0002D 2\$:	SUBW3	#20, 512(R3), (R2)	3528
50		A4	C0 00030	MOVZWL	(R2), STR_PTR	3530
C4 AF	04	60	91 00034	ADDL2	4(R4), STR_PTR	
		04	13 00038	CMPB	(STR_PTR), P.AAS	3532
		62	B7 0003A	BEQL	3\$	
		EF	11 0003C	DECW	(R2)	3535
		04	0003E 3\$:	BRB	2\$	3536
				RET		3539

; Routine Size: 63 bytes, Routine Base: CODE + 1224


```
2616 3540 1 %sbttl 'GET_DIGITAL_LOGO - Create a Phrase of Digital logo'
2617 3541 1 |++
2618 3542 1 | Functional Description:
2619 3543 1 |     VAX/VMS Version Vx.x
2620 3544 1 |
2621 3545 1 | Formal Parameters:
2622 3546 1 |     SCB          - Address of the SCB
2623 3547 1 |     STR_DESC     - Desc of String to Return
2624 3548 1 |     RET_LEN      - Return length of Desc.
2625 3549 1 |
2626 3550 1 | Implicit Inputs:
2627 3551 1 |     none
2628 3552 1 |
2629 3553 1 | Implicit Outputs:
2630 3554 1 |     none
2631 3555 1 |
2632 3556 1 | Returned Value:
2633 3557 1 |     none
2634 3558 1 |
2635 3559 1 | Side Effects:
2636 3560 1 |     none
2637 3561 1 | --
2638 3562 1 ROUTINE GET_DIGITAL_LOGO (
2639 3563 1 |     SCB          : REF $BBLOCK,          ! SCB
2640 3564 1 |     STR_DESC     : REF VECTOR[2],        ! Output buffer desc
2641 3565 1 |     RSL_LEN      : REF VECTOR [,WORD]    ! Return length (word)
2642 3566 1 | ) : NOVALUE =
2643 3567 2 BEGIN
2644 3568 2 BIND
2645 3569 2 | DEFAULT = $DESCRIPTOR (
2646 3570 2 |     '!AC - VAX/VMS Version ',
2647 3571 2 |     '!AS');
2648 3572 2 |
2649 3573 2 LOCAL
2650 3574 2 | LOGO
2651 3575 2 |     FAO_DESC : VECTOR[2],
2652 3576 2 |     BUFFER   : VECTOR[20,byte],
2653 3577 2 |     ITEM_LIST : $ITMLST_DECL (ITEMS=1);
2654 3578 2 |
2655 3579 2 | FAO_DESC[SIZE] = %ALLOCATION(BUFFER);
2656 3580 2 | FAO_DESC[ADDR] = BUFFER;
2657 3581 2 |
2658 3582 2 IF .SCB[PSM$PAGE_WIDTH] LSS 52          ! 52 chars in complete logo
2659 3583 2 THEN
2660 3584 2 | LOGO = UPLIT BYTE (%ASCII 'DEC')
2661 3585 2 |
2662 3586 2 ELSE
2663 3587 2 | LOGO = UPLIT BYTE (%ASCII 'Digital Equipment Corporation');
2664 3588 2 |
2665 3589 2 | $ITMLST_INIT (ITMLST=ITEM_LIST,
2666 3590 2 |     (
2667 3591 2 |         ITMCD=SYIS_VERSION,
2668 3592 2 |         BUFADR=.FAO_DESC[ADDR],
2669 3593 2 |         BUFSIZ=8,
2670 3594 2 |         RETLEN=FAO_DESC[SIZE]
2671 3595 2 |     ));
2672 3596 2 $GETSVIW(ITMLST=ITEM_LIST);
```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_DIGITAL_LOGO - Create a Phrase of Digital

M 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 86
(24)

```

2673 3597 2
2674 3598 2 FAO_DESC[SIZE] = DELIMIT_STRING_NOT (.FAO_DESC[ADDR],
2675 3599 2                                     ^CHAR(32), .FAO_DESC[SIZE]);
2676 3600 2 $FAO (
2677 3601 2     DEFAULT
2678 3602 2     RSL_LEN[0],
2679 3603 2     STR_DESC[0],
2680 3604 2     LOGO,
2681 3605 2     FAO_DESC[0])
2682 3606 2
2683 3607 1 END;

```

! version

```

56 20 53 4D 56 2F 58 41 56 20 2D 20 43 41 21 01263 P.AAU: .ASCII \!AC - VAX/VMS Version \
20 6E 6F 69 73 72 65 01272
53 41 21 01279
00000019 0127C P.AAT: .ASCII \!AS\
00000000 01280 .LONG 25
P.AAV: .ADDRESS P.AAU
P.AAW: .ASCII <3>\DEC\
        .ASCII <29>\Digital Equipment Corporation\
6D 70 69 75 71 45 20 6C 61 74 69 67 45 44 03 01284
6E 6F 69 74 61 72 6F 70 72 6F 43 20 74 6E 65 01288
                                01297

```

DEFAULT= P.AAT
.EXTRN SYS\$GETSYIW

0004 00000 GET_DIGITAL LOGO:

```

24 5E 2C C2 00002 .WORD Save R2 3562
28 AE 14 D0 00005 .SUBL2 #44, SP 3579
AE 10 AE 9E 00009 .MOVL #20, FAO_DESC 3580
50 04 AC D0 0000E .MOVAB BUFFER, FAO_DESC+4 3582
34 0200 C0 D1 00012 .MOVL SCB, R0
06 18 00017 .CMPL 512(R0), #52
52 C2 AF 9E 00019 .BGEQ 1$
04 11 0001D .MOVAB P.AAV, LOGO 3584
52 C0 AF 9E 0001F 1$: .BRB 2$
50 6E 9E 00023 2$: .MOVAB P.AAW, LOGO 3586
80 10000008 8F D0 00026 .MOVAB ITEM_LIST, $$ITMBLKPTR 3594
80 28 AE D0 0002D .MOVL #268735464, ($$ITMBLKPTR)+
80 24 AE 9E 00031 .MOVL FAO_DESC+4, ($$ITMBLKPTR)+
80 D4 00035 .MOVAB FAO_DESC, ($$ITMBLKPTR)+
7E 7C 00037 .CLRL ($$ITMBLKPTR)+
7E D4 00039 .CLRL -(SP) 3596
0C AE 9F 0003B .CLRL -(SP)
7E 7C 0003E .PUSHAB ITEM_LIST
7E D4 00040 .CLRL -(SP)
00000000G 00 07 FB 00042 .CLRL -(SP)
24 AE DD 00049 .CALLS #7, SYS$GETSYIW 3599
20 DD 0004C .PUSHL FAO_DESC 3598
30 AE DD 0004E .PUSHL #32
03 FB 00051 .PUSHL FAO_DESC+4
0000V 24 CF 50 D0 00056 .CALLS #3, DELIMIT_STRING_NOT
24 AE 9F 0005A .MOVL R0, FAO_DESC
08 52 DD 0005D .PUSHAB FAO_DESC 3605
0C AC DD 0005F .PUSHL LOGO
AC DD 00062 .PUSHL STR_DESC
AC DD 00062 .PUSHL RSL_LEN

```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_DIGITAL_LOGO - Create a Phrase of Digital L

N 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 87
(24)

00000000G 00 FF6D CF 9F 00065
05 FB 00069
04 00070

PUSHAB DEFAULT
CALLS #5, SYSSFAO
RET

:
:
: 3607

; Routine Size: 113 bytes, Routine Base: CODE + 12A6

```
2685 3608 1 %sbtll 'GET_JOB_DESCRIPTION - Create a Sentence Describing the Current Job'
2686 3609 1 ++
2687 3610 1 Functional Description:
2688 3611 1 This routine creates a sentence describing the current job.
2689 3612 1
2690 3613 1 Formal Parameters:
2691 3614 1 SCB - Address of the SCB
2692 3615 1 STR_DESC - Desc of String to Return
2693 3616 1 RET_LEN - Return Length of Desc.
2694 3617 1
2695 3618 1 Implicit Inputs:
2696 3619 1 none
2697 3620 1
2698 3621 1 Implicit Outputs:
2699 3622 1 none
2700 3623 1
2701 3624 1 Returned Value:
2702 3625 1 none
2703 3626 1
2704 3627 1 Side Effects:
2705 3628 1 none
2706 3629 1 --
2707 3630 1 ROUTINE GET_JOB_DESCRIPTION (
2708 3631 1 SCB : REF $BLOCK, ! SCB
2709 3632 1 TIME_FLAG : REF VECTOR[2], ! Output buffer desc
2710 3633 1 STR_DESC : REF VECTOR [,WORD] ! Return length (word)
2711 3634 1 RET_LEN : NOVALUE =
2712 3635 1 )
2713 3636 2 BEGIN
2714 3637 2 BIND
2715 3638 2 TRAILING = 1,
2716 3639 2 LEADING = 0,
2717 3640 2
2718 P 3641 2 NODE = $DESCRIPTOR (
2719 3642 2 'SYS$NODE'), ! - system announcement
2720 3643 2
2721 P 3644 2 DATE_FORMAT = $DESCRIPTOR (
2722 3645 2 '!17XD'),
2723 3646 2
2724 P 3647 2 SENT_FORMAT1 = $DESCRIPTOR (
2725 P 3648 2 'Job !AS ', - job name
2726 P 3649 2 '(!UL)', - job number
2727 P 3650 2 'queued to !AS ', - batch file name(pres tense)
2728 P 3651 2 'on !AS ', - time queued
2729 P 3652 2 'by user !AS, ', - user name
2730 P 3653 2 'UIC !XI, ', - user uic
2731 P 3654 2 'under account !AS, ', - user account
2732 P 3655 2 'at priority !UL, ', - que priority
2733 P 3656 2 '!AC ', - 'started'/'completed'/
2734 P 3657 2 'restarted'/'aborted'
2735 P 3658 2 'on printer !AS ', - device name
2736 P 3659 2 'on !AS ', - time printed
2737 P 3660 2 'from queue !AS', - executor queue
2738 3661 2 ' ', - period
2739 3662 2 );
2740 3663 2 LOCAL
2741 3664 2 RET_LENGTH
```



```
2742 3665 2 STATUS
2743 3666 2 DOUBLE_COLONS
2744 3667 2 CHOICE
2745 3668 2 DATE_QUEUED : VECTOR[2],
2746 3669 2 DATE_QUEUED_BUFF : VECTOR [17,byte],
2747 3670 2 DATE_PRINTED : VECTOR[2],
2748 3671 2 DATE_PRINTED_BUFF : VECTOR [17,byte],
2749 3672 2 ACCOUNT_DESC : VECTOR [2],
2750 3673 2 USERNAME_DESC : VECTOR [2]; ! desc of string
2751 3674 2
2752 3675 2 ! get the user name delimited
2753 3676 2
2754 3677 2 USERNAME_DESC[SIZE] = .SCB_SIZE_ (USER_NAME);
2755 3678 2 USERNAME_DESC[ADDR] = .SCB_ADDR_ (USER_NAME);
2756 3679 2 ! Insert only the string ... No trailing blanks
2757 3680 2
2758 3681 2 DISCARD (TRAILING, %C' ', .USERNAME_DESC[ADDR], .USERNAME_DESC[SIZE],
2759 3682 2 USERNAME_DESC[SIZE], USERNAME_DESC[ADDR]); ! Return length and pointer
2760 3683 2
2761 3684 2 ! get the account name delimited
2762 3685 2
2763 3686 2 ACCOUNT_DESC[SIZE] = .SCB_SIZE_ (ACCOUNT_NAME);
2764 3687 2 ACCOUNT_DESC[ADDR] = .SCB_ADDR_ (ACCOUNT_NAME);
2765 3688 2 ! Insert only the string ... No trailing blanks
2766 3689 2
2767 3690 2 DISCARD (TRAILING, %C' ', .ACCOUNT_DESC[ADDR], .ACCOUNT_DESC[SIZE],
2768 3691 2 ACCOUNT_DESC[SIZE], ACCOUNT_DESC[ADDR]); ! Return length and pointer
2769 3692 2
2770 3693 2 ! start, restart and complete
2771 3694 2 IF .TIME_FLAG
2772 3695 2 THEN
2773 3696 2 BEGIN
2774 3697 2 CHOICE = UPLIT BYTE (%ASCIC 'started');
2775 3698 2 IF .REQUEST_FLAG_ (RESTARTING)
2776 3699 2 THEN
2777 3700 2 CHOICE = UPLIT BYTE (%ASCIC 'restarted');
2778 3701 2 END
2779 3702 2 ELSE
2780 3703 2 BEGIN
2781 3704 2 RIND CONDITION = SCB[PSMST_CONDITION_AREA] : VECTOR; ! Task completion status
2782 3705 2
2783 3706 2 ! Assume job completed normally
2784 3707 2
2785 3708 2 CHOICE = UPLIT BYTE (%ASCIC 'completed');
2786 3709 2
2787 3710 2 ! Check completion status for an error
2788 3711 2
2789 3712 2 IF .CONDITION[0] NEQU 0
2790 3713 2 THEN
2791 3714 2 BEGIN
2792 3715 2 ! Assume job controller or symbiont initiated abort
2793 3716 2
2794 3717 2 CHOICE = UPLIT BYTE (%ASCIC 'ABORTED');
2795 3718 2
2796 3719 2 ! Check for special case of job controller initiated requeue
2797 3720 2
2798 3721 2
```

```
2799 3722 4 !*! FEATURE DISABLED UNTIL JOB CONTROLLER MESSAGE AVAILABILITY STRAIGHTENED OUT
2800 3723 4 !*! IF .CONDITION[0] EQLU JBC$_JOBQUEUE
2801 3724 4 !*! THEN
2802 3725 4 !*! CHOICE = UPLIT BYTE (%ASCIC 'REQUEUED');
2803 3726 4
2804 3727 3 END;
2805 3728 2 END;
2806 3729 2
2807 3730 2 ! Get and delimit the date/times
2808 3731 2 ! time queued
2809 3732 2 DATE_QUEUED[SIZE] = %ALLOCATION(DATE_QUEUED_BUFF);
2810 3733 2 DATE_QUEUED[ADDR] = DATE_QUEUED_BUFF;
2811 3734 2
2812 P 3735 2 $FAO ( DATE_FORMAT,
2813 P 3736 2 RET_LENGTH,
2814 P 3737 2 DATE_QUEUED[0],
2815 3738 2 SCB[PSM$Q_TIME_QUEUED]);
2816 3739 2
2817 3740 2 DISCARD (LEADING, %C' ', DATE_QUEUED[ADDR], RET_LENGTH,
2818 3741 2 DATE_QUEUED[SIZE], DATE_QUEUED[ADDR]); ! Return length and pointer
2819 3742 2 DISCARD (TRAILING, %C' ', DATE_QUEUED[ADDR], RET_LENGTH,
2820 3743 2 DATE_QUEUED[SIZE], DATE_QUEUED[ADDR]); ! Return length and pointer
2821 3744 2
2822 3745 2 ! time printed
2823 3746 2 DATE_PRINTED[SIZE] = %ALLOCATION(DATE_PRINTED_BUFF);
2824 3747 2 DATE_PRINTED[ADDR] = DATE_PRINTED_BUFF;
2825 3748 2
2826 P 3749 2 $FAO ( DATE_FORMAT,
2827 P 3750 2 RET_LENGTH,
2828 P 3751 2 DATE_PRINTED[0],
2829 3752 2 SCB[PSM$Q_TIME_PRINTED]);
2830 3753 2
2831 3754 2 DISCARD (LEADING, %C' ', DATE_PRINTED[ADDR], RET_LENGTH,
2832 3755 2 DATE_PRINTED[SIZE], DATE_PRINTED[ADDR]); ! Return length and pointer
2833 3756 2 DISCARD (TRAILING, %C' ', DATE_PRINTED[ADDR], RET_LENGTH,
2834 3757 2 DATE_PRINTED[SIZE], DATE_PRINTED[ADDR]); ! Return length and pointer
2835 3758 2
2836 P 3759 2 $FAO ( SENT_FORMAT1,
2837 P 3760 2 RET_LEN[0],
2838 P 3761 2 STR_DESC[0],
2839 P 3762 2 SCB[PSM$Q_JOB_NAME], ! job name
2840 P 3763 2 .SCB[PSM$Q_ENTRY_NUMBER], ! entry number
2841 P 3764 2 SCB[PSM$Q_QUEUE], ! batch que present tense
2842 P 3765 2 DATE_QUEUED[0], ! time queued
2843 P 3766 2 USERNAME_DESC[0], ! user name
2844 P 3767 2 .SCB[PSM$Q_UIC], ! user uic
2845 P 3768 2 ACCOUNT_DESC[0], ! user account
2846 P 3769 2 .SCB[PSM$Q_PRIORITY], ! queue priority
2847 P 3770 2 .CHOICE, ! started/completed/restarted
2848 P 3771 2 SCB[PSM$Q_DEVICE_NAME], ! device name
2849 P 3772 2 DATE_PRINTED[0], ! time printed
2850 P 3773 2 SCB[PSM$Q_EXECUTOR_QUEUE] ! executor queue
2851 3774 2 );
2852 3775 2
2853 3776 2 RETURN SSS_NORMAL;
2854 3777 1 END;
```

	45	44	4F	4E	24	53	59	53	01317	P.AAY:	.ASCII	\SYS\$NODE\
									0131F		.BLKB	1
								00000008	01320	P.AAX:	.LONG	8
								00000000	01324		.ADDRESS	P.AAY
				44	25	37	31	21	01328	P.ABA:	.ASCII	\!17XD\
									0132D		.BLKB	3
								00000005	01330	P.AAZ:	.LONG	5
								00000000	01334		.ADDRESS	P.ABA
	20	53	41	21	20	62	6F	4A	01338	P.ABC:	.ASCII	\Job !AS \
									01340		.ASCII	\(!UL) \
20	53	41	21	20	6F	74	20	64	01346		.ASCII	\queued to !AS \
									01354		.ASCII	\on !AS \
	20	2C	53	41	21	20	72	65	0135B		.ASCII	\by user !AS, \
									01368		.ASCII	\UIC !XI, \
21	20	74	6E	75	6F	63	63	61	01371		.ASCII	\under account !AS \
									01380			
4C	55	21	20	79	74	69	72	6F	01383		.ASCII	\at priority !UL, \
									01392			
									01394		.ASCII	\!AC \
20	53	41	21	20	72	65	74	6E	01398		.ASCII	\on printer !AS \
									013A7		.ASCII	\on !AS \
	53	41	21	20	65	75	65	75	013AE		.ASCII	\from queue !AS\
									013BC		.ASCII	\.\
									013BD		.BLKB	3
									00000085	013C0	P.ABB:	.LONG 133
									00000000	013C4		.ADDRESS P.ABC
									013C8	P.ABD:	.ASCII	<7>\started\
									013D0	P.ABE:	.ASCII	<9>\restarted\
									013DA	P.ABF:	.ASCII	<9>\completed\
									013E4	P.ABG:	.ASCII	<7>\ABORTED\

```
TRAILING= 1
LEADING= 0
NODE= P.AAX
DATE-FORMAT= P.AAZ
SENT-FORMAT1= P.ABB
```

PC	Instruction	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
----	-------------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_JOB_DESCRIPTION - Create a Sentence

F 4
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 92
(25)

OC	AE	60	3C	0003B	MOVZWL	(R0), ACCOUNT_DESC	
10	AE	04	A0	DD 0003F	MOVL	4(R0), ACCOUNT_DESC+4	3687
		10	AE	9F 00044	PUSHAB	ACCOUNT_DESC+4	3691
		10	AE	9F 00047	PUSHAB	ACCOUNT_DESC	
		14	AE	DD 0004A	PUSHL	ACCOUNT_DESC	3690
		1C	AE	DD 0004D	PUSHL	ACCOUNT_DESC+4	
		20	DD	00050	PUSHL	#32	
		01	DD	00052	PUSHL	#1	
	65	06	FB	00054	CALLS	#6, DISCARD	
	12	08	AC	E9 00057	BLBC	TIME FLAG, 1\$	3694
	53	FF7D	CF	9E 0005B	MOVAB	P.ABF, CHOICE	3697
17	0140	C2	02	E1 00060	BBC	#2, 320(R2), 2\$	3698
		53	FF7A	CF 9E 00066	MOVAB	P.ABE, CHOICE	3700
			10	11 0006B	BRB	2\$	3694
		53	FF7D	CF 9E 0006D	MOVAB	P.ABF, CHOICE	3708
			028E	C2 D5 00072	TSTL	654(R2)	3712
			05	13 00076	BEQL	2\$	
	53	FF7C	CF	9E 00078	MOVAB	P.ABG, CHOICE	3717
44	AE		11	DD 0007D	MOVL	#17, DATE_QUEUED	3732
48	AE	30	AE	9E 00081	MOVAB	DATE_QUEUED_BUFF, DATE_QUEUED+4	3733
		015C	C2	9F 00086	PUSHAB	348(R2)	3738
		48	AE	9F 0008A	PUSHAB	DATE_QUEUED	
		08	AE	9F 0008D	PUSHAB	RET LENGTH	
		FEB0	CF	9F 00090	PUSHAB	DATE_FORMAT	
	64		04	FB 00094	CALLS	#4, SYSSFAO	
		48	AE	9F 00097	PUSHAB	DATE_QUEUED+4	3741
		48	AE	9F 0009A	PUSHAB	DATE_QUEUED	
		08	AE	DD 0009D	PUSHL	RET LENGTH	3740
		54	AE	DD 000A0	PUSHL	DATE_QUEUED+4	
		20	DD	000A3	PUSHL	#32	
		7E	D4	000A5	CLRL	-(SP)	
	65		06	FB 000A7	CALLS	#6, DISCARD	
		48	AE	9F 000AA	PUSHAB	DATE_QUEUED+4	3743
		48	AE	9F 000AD	PUSHAB	DATE_QUEUED	
		08	AE	DD 000B0	PUSHL	RET LENGTH	3742
		54	AE	DD 000B3	PUSHL	DATE_QUEUED+4	
		20	DD	000B6	PUSHL	#32	
		01	DD	000B8	PUSHL	#1	
		06	FB	000BA	CALLS	#6, DISCARD	
28	AE		11	DD 000BD	MOVL	#17, DATE_PRINTED	3746
2C	AE	14	AE	9E 000C1	MOVAB	DATE_PRINTED_BUFF, DATE_PRINTED+4	3747
		0234	C2	9F 000C6	PUSHAB	564(R2)	3752
		2C	AE	9F 000CA	PUSHAB	DATE_PRINTED	
		08	AE	9F 000CD	PUSHAB	RET LENGTH	
		FE70	CF	9F 000D0	PUSHAB	DATE_FORMAT	
	64		04	FB 000D4	CALLS	#4, SYSSFAO	
		2C	AE	9F 000D7	PUSHAB	DATE_PRINTED+4	3755
		2C	AE	9F 000DA	PUSHAB	DATE_PRINTED	
		08	AE	DD 000DD	PUSHL	RET LENGTH	3754
		38	AE	DD 000E0	PUSHL	DATE_PRINTED+4	
		20	DD	000E3	PUSHL	#32	
		7E	D4	000E5	CLRL	-(SP)	
	65		06	FB 000E7	CALLS	#6, DISCARD	
		2C	AE	9F 000EA	PUSHAB	DATE_PRINTED+4	3757
		2C	AE	9F 000ED	PUSHAB	DATE_PRINTED	
		08	AE	DD 000F0	PUSHL	RET LENGTH	3756
		38	AE	DD 000F3	PUSHL	DATE_PRINTED+4	

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_JOB_DESCRIPTION - Create a Sentence Describ

G 4
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 93
(25)

	20	DD	000F6	PUSHL	#32
	01	DD	000F8	PUSHL	#1
65	06	FB	000FA	CALLS	#6, DISCARD
	5C	A2	9F 000FD	PUSHAB	92(R2)
	2C	AE	9F 00100	PUSHAB	DATE_PRINTED
	4C	A2	9F 00103	PUSHAB	76(R2)
	53	DD	00106	PUSHL	CHOICE
0128	C2	DD	00108	PUSHL	296(R2)
20	AE	9F	0010C	PUSHAB	ACCOUNT_DESC
0168	C2	DD	0010F	PUSHL	360(R2)
20	AE	9F	00113	PUSHAB	USERNAME_DESC
64	AE	9F	00116	PUSHAB	DATE_QUEUED
012C	C2	9F	00119	PUSHAB	300(R2)
58	A2	DD	0011D	PUSHL	88(R2)
00A8	C2	9F	00120	PUSHAB	168(R2)
0C	AC	DD	00124	PUSHL	STR_DESC
10	AC	DD	00127	PUSHL	RET_LEN
FEA6	CF	9F	0012A	PUSHAB	SENT_FORMAT1
64	0F	FB	0012E	CALLS	#15, -SYSSFAO
	04		00131	RET	

3774

3777

; Routine Size: 306 bytes. Routine Base: CODE + 13EC

```
2856 3778 1 %sbttl 'GET_FILE_DESCRIPTION - Create a Sentence Describing the Current File'
2857 3779 1 ++
2858 3780 1 Functional Description:
2859 3781 1 This routine creates a sentence describing the current File.
2860 3782 1
2861 3783 1 Formal Parameters:
2862 3784 1 SCB - Address of the SCB
2863 3785 1 STR_DESC - Desc of String to Return
2864 3786 1 RET_LEN - Return length of Desc.
2865 3787 1
2866 3788 1 Implicit Inputs:
2867 3789 1 none
2868 3790 1
2869 3791 1 Implicit Outputs:
2870 3792 1 none
2871 3793 1
2872 3794 1 Returned Value:
2873 3795 1 none
2874 3796 1
2875 3797 1 Side Effects:
2876 3798 1 none
2877 3799 1 --
2878 3800 1 ROUTINE GET_FILE_DESCRIPTION (
2879 3801 1 SCB : REF $BLOCK, : SCB
2880 3802 1 STR_DESC : REF VECTOR[2], : Output buffer desc
2881 3803 1 RET_LEN : REF VECTOR [,WORD] : Return length (word)
2882 3804 1 ) : NOVALUE =
2883 3805 2 BEGIN
2884 3806 2 BIND
2885 3807 2 FAB = .SCB[PSMSA_FAB]: $BLOCK,
2886 3808 2 NAM = .SCB[PSMSA_NAM]: $BLOCK,
2887 3809 2 XABDAT = .SCB[PSMSA_XABDAT]: $BLOCK,
2888 3810 2 XABFHC = .SCB[PSMSA_XABFHC]: $BLOCK,
2889 3811 2 XABPRO = .SCB[PSMSA_XABPRO]: $BLOCK,
2890 3812 2
2891 P 3813 2 FORMAT_POS = $DESCRIPTOR (
2892 P 3814 2 'File !AS ' : - file name
2893 P 3815 2 '(!UL,!UL,!UL), ' : - file Id number
2894 P 3816 2 'last revised on !17XD, ' : - revision date
2895 P 3817 2 'is a !UL block ' : - file size
2896 P 3818 2 '!AC file ' : - file organization
2897 3819 2 'owned by UIC !XI. '), : - owner user uic
2898 3820 2
2899 P 3821 2 RECORD_FORMAT = $DESCRIPTOR (
2900 P 3822 2 'The records are ' : -
2901 3823 2 '!AC with '), : - record format
2902 3824 2
2903 P 3825 2 RECORD_VFC_FORMAT = $DESCRIPTOR (
2904 P 3826 2 'The records are ' : -
2905 P 3827 2 'variable length with a ' : -
2906 3828 2 'fixed control size of !UL byte!XS and '), : - fixed control area size
2907 3829 2
2908 3830 2
2909 P 3831 2 REC_SIZE = $DESCRIPTOR (
2910 3832 2 'The longest record is !UL byte!XS.'), : - max record size
2911 3833 2
2912 3834 2
```

```
2913 P 3835 2 CARRIAGE_FORMAT = $DESCRIPTOR (
2914 3836 '!'AC'), ! - record attributes
2915 3837
2916 P 3838 FORMAT_NEG = $DESCRIPTOR (
2917 3839 'File (!AS) description is unavailable to the symbiont.');
```

```
2918 3840
2919 3841 LITERAL
2920 3842 K_MAX_BUFFER_SIZE = 512;
2921 3843
2922 3844 LOCAL
2923 3845 RECORD_SIZE ,
2924 3846 FILE_SIZE ,
2925 3847 ORGANIZATION ,
2926 3848 ATTRIBUTES ,
2927 3849 FORMAT ,
2928 3850 CURRENT_LEN : INITIAL (0),
2929 3851 DATE_REVISD : VECTOR[2],
2930 3852 STRING_PTR : VECTOR [2]; ! Pointer to current string
2931 3853
2932 3854 ! Allocate the buffer for "GET_xxx" Routines
2933 3855
2934 3856 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
2935 3857 STRING_PTR[ADDR] = .STR_DESC[ADDR]; ! init address
2936 3858
2937 3859 RET_LEN[0] = 0;
2938 3860
2939 3861 IF FILE_OPEN(.SCB)
2940 3862 THEN
2941 3863 BEGIN
2942 3864 ! get the file size
2943 3865
2944 3866 FILE_SIZE = .XABFHC[XAB$L_EBK];
2945 3867 IF (.XABFHC[XAB$W_FFB] EQL 0) AND
2946 3868 (.FILE_SIZE NEQ 0)
2947 3869 THEN
2948 3870 FILE_SIZE = .FILE_SIZE - 1;
2949 3871
2950 3872 ! insert file organization
2951 3873
2952 3874 IF .FAB[FAB$B_ORG] EQL FAB$C_IDX
2953 3875 THEN
2954 3876 ORGANIZATION = UPLIT BYTE (XASCIC 'indexed')
2955 3877 ELSE IF .FAB[FAB$B_ORG] EQL FAB$C_SEQ
2956 3878 THEN
2957 3879 ORGANIZATION = UPLIT BYTE (XASCIC 'sequential')
2958 3880 ELSE IF .FAB[FAB$B_ORG] EQL FAB$C_REL
2959 3881 THEN
2960 3882 ORGANIZATION = UPLIT BYTE (XASCIC 'relative')
2961 3883 ELSE
2962 3884 ORGANIZATION = UPLIT BYTE (XASCIC 'undefined organization');
2963 3885
2964 3886
2965 P 3887 $FAO ( FORMAT_POS,
2966 PP 3888 CURRENT_LEN,
2967 PP 3889 STRING_PTR[0],
2968 PP 3890 SCB[PSM$Q_FILE_SPECIFICATION],
2969 P 3891 .NAM[NAM$Q_FID_NUM],
```

```
2970 P 3892 .NAM[NAM$W_FID_SEQ],
2971 P 3893 .NAM[NAM$W_FID_RVN],
2972 P 3894 XABDAT[XAB$Q_RDT],
2973 P 3895 .FILE_SIZE,
2974 P 3896 .ORGANIZATION,
2975 P 3897 .XABPRO[XAB$L_UIC]
2976 );
2977
2978 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
2979 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
2980 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
2981
2982 ! get record format
2983 !
2984
2985 RECORD_SIZE = .XABFHC[XAB$W_LRL]; ! record size
2986
2987 IF .FAB[FAB$B_RFM] NEQ FAB$C_VFC
2988 THEN
2989 BEGIN
2990 ! get record type
2991 !
2992 SELECTONE .FAB[FAB$B_RFM] OF
2993 SET
2994 [FAB$C_FIX]: FORMAT = UPLIT BYTE
2995 (XASCIC 'fixed-length');
2996 [FAB$C_STM]: FORMAT = UPLIT BYTE
2997 (XASCIC 'stream');
2998 [FAB$C_STMCR]: FORMAT = UPLIT BYTE
2999 (XASCIC 'stream-CR');
3000 [FAB$C_STMLF]: FORMAT = UPLIT BYTE
3001 (XASCIC 'stream-LF');
3002 [FAB$C_UDF]: FORMAT = UPLIT BYTE
3003 (XASCIC 'an undefined format');
3004 [FAB$C_VAR]: FORMAT = UPLIT BYTE
3005 (XASCIC 'variable length');
3006 TES;
3007
3008 P 3930 $FAO ( RECORD FORMAT,
3009 P 3931 CURRENT_LEN,
3010 P 3932 STRING_PTR[0],
3011 P 3933 .FORMAT
3012 );
3013
3014 END
3015 ELSE
3016 P 3938 $FAO ( RECORD VFC FORMAT,
3017 P 3939 CURRENT_LEN,
3018 P 3940 STRING_PTR[0],
3019 P 3941 .FAB[FAB$B_FSZ]
3020 );
3021
3022 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3023 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3024 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3025
3026 ! get carriage control
```



```
3027 3949 !
3028 3950 IF .FAB[FAB$V_CR]
3029 3951 THEN
3030 3952     ATTRIBUTES = UPLIT BYTE (%ASCIC 'implied (CR) carriage control')
3031 3953 ELSE
3032 3954     BEGIN
3033 3955     IF .FAB[FAB$V_FTN]
3034 3956     THEN
3035 3957         ATTRIBUTES = UPLIT BYTE (%ASCIC 'FORTRAN (FTN) carriage control')
3036 3958     ELSE
3037 3959         BEGIN
3038 3960         IF .FAB[FAB$V_PRN]
3039 3961         THEN
3040 3962             ATTRIBUTES =
3041 3963                 UPLIT BYTE (%ASCIC 'print file (PRN) carriage control')
3042 3964         ELSE
3043 3965             ATTRIBUTES = UPLIT BYTE
3044 3966                 (%ASCIC 'imbedded (<none>) carriage control');
3045 3967         END;
3046 3968     END;
3047 3969
3048 3970 $FAO (
3049 3971     CARRIAGE FORMAT,
3050 3972     CURRENT_LEN,
3051 3973     STRING_PTR[0],
3052 3974     .ATTRIBUTES );
3053 3975
3054 3976 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3055 3977 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3056 3978 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3057 3979
3058 3980 IF .FAB[FAB$B_RFM] NEQ FAB$C_FIX
3059 3981 THEN
3060 3982     BEGIN
3061 3983     $FAO (
3062 3984         REC_SIZE,
3063 3985         CURRENT_LEN,
3064 3986         STRING_PTR[0],
3065 3987         .RECORD_SIZE );
3066 3988
3067 3989     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3068 3990     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3069 3991     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3070 3992     END;
3071 3993
3072 3994 END
3073 3995 ELSE
3074 3996     BEGIN
3075 3997     $FAO (
3076 3998         FORMAT NEG,
3077 3999         CURRENT_LEN,
3078 4000         STRING_PTR[0],
3079 4001         SCB[PSM$Q_FILE_SPECIFICATION]
3080 4002     );
3081 4003
3082 4004     RET_LEN[0] = .CURRENT_LEN;
3083 4005     END;
```

```

: 3084      4006 2
: 3085      4007 2 ! final check for overflow
: 3086      4008 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 3087      4009 2 THEN
: 3088      4010 2     BEGIN
: 3089      4011 2         RET_LEN[0] = 512;
: 3090      4012 2         RETURN;
: 3091      4013 2     END;
: 3092      4014 2
: 3093      4015 2 RETURN SSS_NORMAL;
: 3094      4016 2 END;
```

```

20 2C 29 4C 55 21 20 53 41 21 20 65 6C 69 46 0151E P.ABI: .ASCII \File !AS \
6E 6F 20 64 65 73 69 76 65 72 20 74 73 61 6C 01527 .ASCII \(!UL,!UL,!UL), \
: 3092      4014 2
: 3093      4015 2 RETURN SSS_NORMAL;
: 3094      4016 2 END;
20 6B 63 6F 6C 62 20 4C 55 21 20 61 20 73 69 01536 .ASCII \last revised on !17%D, \
25 21 20 43 49 55 20 79 62 20 64 65 6E 77 6F 01545 .ASCII \is a !UL block \
: 3092      4014 2
: 3093      4015 2 RETURN SSS_NORMAL;
: 3094      4016 2 END;
65 72 61 20 73 64 72 6F 63 65 72 20 65 68 54 0154D .ASCII \!AC file \
20 68 74 69 77 20 43 41 21 0155C .ASCII \owned by UIC !%I. \
: 3092      4014 2
: 3093      4015 2 RETURN SSS_NORMAL;
: 3094      4016 2 END;
65 72 61 20 73 64 72 6F 63 65 72 20 65 68 54 01574 .LONG 90
68 74 67 6E 65 6C 20 65 6C 62 61 69 72 61 76 01578 P.ABH: .ADDRESS P.ABI
73 20 6C 6F 72 74 6E 6F 63 20 64 65 78 69 66 0157C .ASCII \The records are \
65 74 79 62 20 4C 55 21 20 66 6F 20 65 7A 69 01580 P.ABK: .ASCII \!AC with \
2E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 0158F .BLKB 3
: 3092      4014 2
: 3093      4015 2 RETURN SSS_NORMAL;
: 3094      4016 2 END;
63 73 65 64 20 29 53 41 21 28 20 65 6C 69 46 01590 P.ABJ: .LONG 25
76 61 6E 75 20 73 69 20 6E 6F 69 74 70 69 72 0159C .ADDRESS P.ABK
2E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 015A0 P.ABM: .ASCII \The records are \
6C 61 69 64 65 78 69 66 015A4 .ASCII \variable length with a \
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 015B3 .ASCII \fixed control size of !UL byte!%S and \
6E 6F 69 62 6D 79 73 20 65 68 74 20 015B4 .BLKB 3
: 3092      4014 2
: 3093      4015 2 RETURN SSS_NORMAL;
: 3094      4016 2 END;
63 73 65 64 20 29 53 41 21 28 20 65 6C 69 46 015F1 P.ABL: .LONG 77
76 61 6E 75 20 73 69 20 6E 6F 69 74 70 69 72 015F4 .ADDRESS P.ABM
2E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 015F8 P.ABO: .ASCII \ The longest record is !UL byte!%S.\
6C 61 69 64 65 78 65 64 6E 69 07 0160B .LONG 36
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 0161A .ADDRESS P.ABO
6C 61 69 64 65 78 65 64 6E 69 07 01620 P.ABN: .ASCII \!AC.\
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 01624 P.ABQ: .LONG 4
6C 61 69 64 65 78 65 64 6E 69 07 01628 P.ABP: .ADDRESS P.ABQ
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 01630 P.ABS: .ASCII \File (!AS) description is unavailable to\
6C 61 69 64 65 78 65 64 6E 69 07 01634 .BLKB 2
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 01643 P.ABR: .LONG 54
6C 61 69 64 65 78 65 64 6E 69 07 01652 .ADDRESS P.ABS
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 0165C P.ABT: .ASCII <7>\indexed\
6C 61 69 64 65 78 65 64 6E 69 07 0166A P.ABU: .ASCII <10>\sequential\
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 01670
6C 61 69 64 65 78 65 64 6E 69 07 01674
7E 74 6E 6F 69 62 6D 79 73 20 65 68 74 20 0167C
```

SEPARATE
V04-001

Print Symbiont -- separation routines

GET_FILE_DESCRIPTION - Create a Sentence Descri

M 4

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742

[PRTSMB.SRC]SEPARATE.B32;2

Page 99

(26)

61	67	72	6F	20	64	65	76	69	74	61	6C	65	72	08	01687	P.ABV:	.ASCII	<8>\relative\
						6E	6E	69	66	65	64	6E	75	16	01690	P.ABW:	.ASCII	<22>\undefined organization\
		68	74	67	6E	65	6C	2D	64	65	78	69	66	0C	0169F			
								6D	61	65	72	74	73	06	016A7	P.ABX:	.ASCII	<12>\fixed-length\
					52	43	2D	6D	61	65	72	74	73	09	016B4	P.ABY:	.ASCII	<6>\stream\
					46	4C	2D	6D	61	65	72	74	73	09	016BB	P.ABZ:	.ASCII	<9>\stream-CR\
66	20	64	65	6E	69	66	65	64	6E	75	20	6E	61	13	016C5	P.ACA:	.ASCII	<9>\stream-LF\
										74	61	6D	72	6F	016CF	P.ACB:	.ASCII	<19>\an undefined format\
74	67	6E	65	6C	20	65	6C	62	61	69	72	61	76	0F	016DE			
														68	016E3	P.ACC:	.ASCII	<15>\variable length\
														1D	016F2			
63	20	29	52	43	28	20	64	65	69	6C	70	6D	69	1D	016F3	P.ACD:	.ASCII	<29>\implied (CR) carriage control\
6C	6F	72	74	6E	6F	63	20	65	67	61	69	72	72	61	01702			
20	29	4E	54	46	28	20	4E	41	52	54	52	4F	46	1E	01711	P.ACE:	.ASCII	<30>\FORTRAN (FTN) carriage control\
6F	72	74	6E	6F	63	20	65	67	61	69	72	72	61	63	01720			
														6C	0172F			
52	50	28	20	65	6C	69	66	20	74	6E	69	72	70	21	01730	P.ACF:	.ASCII	\\!print file (PRN) carriage control\
6E	6F	63	20	65	67	61	69	72	72	61	63	20	29	4E	0173F			
											6C	6F	72	74	0174E			
6E	6F	6E	3C	28	20	64	65	64	64	65	62	6D	69	22	01752	P.ACG:	.ASCII	\\'imbedded (<none>) carriage control\
6F	63	20	65	67	61	69	72	72	61	63	20	29	3E	65	01761			
										6C	6F	72	74	6E	01770			

FORMAT_POS= P.ABH
RECORD_FORMAT= P.ABJ
RECORD_VFC_FORMAT= P.ABL
REC_SIZE= P.ABN
CARRIAGE_FORMAT= P.ABP
FORMAT_NEG= P.ABR

07FC 00000 GET_FILE_DESCRIPTION:

5A	00000000G	00	9E	00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10	3800
59	FEF2	CF	9E	00009	MOVAB	SYSSFA0, R10	
5E		10	C2	0000E	MOVAB	P.ABT, R9	
50	04	AC	7D	00011	SUBL2	#16, SP	
52	0248	CO	DO	00015	MOVQ	SCB, R0	3807
54	024C	CO	DO	0001A	MOVL	584(R0), R2	
57	0254	CO	DO	0001F	MOVL	588(R0), R4	3808
53	0258	CO	DO	00024	MOVL	596(R0), R7	3809
56	025C	CO	DO	00029	MOVL	600(R0), R3	3810
					MOVL	604(R0), R6	3811
		7E	D4	0002E	CLRL	CURRENT_LEN	
04	AE	0200	8F	3C	MOVZWL	#512, STRING_PTR	3856
08	AE	04	A1	DO	MOVL	4(R1), STRING_PTR+4	3857
55	0C	AC	DO	0003B	MOVL	RET_LEN, R5	3859
		65	B4	0003F	CLRW	(R5)	
58	0098	CO	9E	00041	MOVAB	152(R0), R8	3898
		50	DD	00046	PUSHL	R0	3861
0000V	CF	01	FB	00048	CALLS	#1, FILE_OPEN	
03		50	E8	0004D	BLBS	R0, 1\$	
		0162	31	00050	BRW	19\$	
51	10	A3	DO	00053	MOVL	16(R3), FILE_SIZE	3866
	14	A3	B5	00057	TSTW	20(R3)	3867
		06	12	0005A	BNEQ	2\$	
		51	D5	0005C	TSTL	FILE_SIZE	3868
		02	13	0005E	BEQL	2\$	

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_FILE_DESCRIPTION - Create a Sentence

N 4
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 100
(26)

		51	D7	00060	DECL	FILE SIZE	3870
20	1D	A2	91	00062	2\$: CMPB	29(R2), #32	3874
		05	12	00066	BNEQ	3\$	
50		69	9E	00068	MOVAB	P.ABT, ORGANIZATION	3876
		1B	11	0006B	BRB	6\$	
	1D	A2	95	0006D	3\$: TSTB	29(R2)	3877
		06	12	00070	BNEQ	4\$	
50	08	A9	9E	00072	MOVAB	P.ABU, ORGANIZATION	3879
		10	11	00076	BRB	6\$	
10	1D	A2	91	00078	4\$: CMPB	29(R2), #16	3880
		06	12	0007C	BNEQ	5\$	
50	13	A9	9E	0007E	MOVAB	P.ABV, ORGANIZATION	3882
		04	11	00082	BRB	6\$	
50	1C	A9	9E	00084	5\$: MOVAB	P.ABW, ORGANIZATION	3884
	0C	A6	DD	00088	6\$: PUSHL	12(R6)	3898
		50	DD	0008B	PUSHL	ORGANIZATION	
		51	DD	0008D	PUSHL	FILE SIZE	
	0C	A7	9F	0008F	PUSHAB	12(R7)	
7E	28	A4	3C	00092	MOVZWL	40(R4), -(SP)	
7E	26	A4	3C	00096	MOVZWL	38(R4), -(SP)	
7E	24	A4	3C	0009A	MOVZWL	36(R4), -(SP)	
		58	DD	0009E	PUSHL	R8	
	24	AE	9F	000A0	PUSHAB	STRING_PTR	
	24	AE	9F	000A3	PUSHAB	CURRENT_LEN	
	FF04	C9	9F	000A6	PUSHAB	FORMAT_POS	
6A		0B	FB	000AA	CALLS	#11, SYSS\$FA0	
65		6E	A0	000AD	ADDW2	CURRENT_LEN, (R5)	3900
08	AE	6E	C0	000B0	ADDL2	CURRENT_LEN, STRING_PTR+4	3901
04	AE	65	3C	000B4	MOVZWL	(R5), STRING_PTR	3902
BF	04	AE	C3	000B8	SUBL3	STRING_PTR, #512, STRING_PTR	
54	0A	A3	3C	000C2	MOVZWL	10(R3), RECORD_SIZE	3907
53	1F	A2	9A	000C6	MOVZBL	31(R2), R3	3909
03		53	91	000CA	CMPB	R3, #3	
		4D	13	000CD	BEQL	13\$	
01		53	91	000CF	CMPB	R3, #1	3916
		06	12	000D2	BNEQ	7\$	
50	33	A9	9E	000D4	MOVAB	P.ABX, FORMAT	
		34	11	000D8	BRB	12\$	
04		53	91	000DA	7\$: CMPB	R3, #4	3918
		06	12	000DD	BNEQ	8\$	
50	40	A9	9E	000DF	MOVAB	P.ABY, FORMAT	
		29	11	000E3	BRB	12\$	
06		53	91	000E5	8\$: CMPB	R3, #6	3920
		06	12	000E8	BNEQ	9\$	
50	47	A9	9E	000EA	MOVAB	P.ABZ, FORMAT	
		1E	11	000EE	BRB	12\$	
05		53	91	000F0	9\$: CMPB	R3, #5	3922
		06	12	000F3	BNEQ	10\$	
50	51	A9	9E	000F5	MOVAB	P.ACA, FORMAT	
		13	11	000F9	BRB	12\$	
		53	D5	000FB	10\$: TSTL	R3	3924
		06	12	000FD	BNEQ	11\$	
50	5B	A9	9E	000FF	MOVAB	P.ACB, FORMAT	
		09	11	00103	BRB	12\$	
02		53	91	00105	11\$: CMPB	R3, #2	3926
		04	12	00108	BNEQ	12\$	
50	6F	A9	9E	0010A	MOVAB	P.ACC, FORMAT	

				08	50	DD	0010E	12\$:	PUSHL	FORMAT	3934	
				08	AE	9F	00110		PUSHAB	STRING_PTR		
				08	AE	9F	00113		PUSHAB	CURRENT_LEN		
				FF28	C9	9F	00116		PUSHAB	RECORD_FORMAT		
					0D	11	0011A		BRB	14\$		
		7E		3F	A2	9A	0011C	13\$:	MOVZBL	63(R2), -(SP)	3942	
				08	AE	9F	00120		PUSHAB	STRING_PTR		
				08	AE	9F	00123		PUSHAB	CURRENT_LEN		
				80	A9	9F	00126		PUSHAB	RECORD_VFC_FORMAT		
		6A			04	FB	00129	14\$:	CALLS	#4, SYSSFAO		
		65			6E	A0	0012C		ADDW2	CURRENT_LEN, (R5)	3944	
		08			6E	C0	0012F		ADDL2	CURRENT_LEN, STRING_PTR+4	3945	
		04			65	3C	00133		MOVZWL	(R5), STRING_PTR	3946	
04	AE	00000200		8F	04	AE	C3	00137	SUBL3	STRING_PTR, #512, STRING_PTR		
	06	1E		A2	01	E1	00141		BBC	#1, 30(R2), 15\$	3950	
				50	7F	A9	9E	00146	MOVAB	P.ACD, ATTRIBUTES	3952	
					1C	11	0014A		BRB	18\$		
		07		1E	A2	E9	0014C	15\$:	BLBC	30(R2), 16\$	3955	
		50		009D	C9	9E	00150		MOVAB	P.ACE, ATTRIBUTES	3957	
					11	11	00155		BRB	18\$		
	07	1E		A2	02	E1	00157	16\$:	BBC	#2, 30(R2), 17\$	3960	
				50	00BC	C9	9E	0015C	MOVAB	P.ACF, ATTRIBUTES	3963	
					05	11	00161		BRB	18\$	3962	
				50	00DE	C9	9E	00163	MOVAB	P.ACG, ATTRIBUTES	3965	
						50	DD	00168	18\$:	PUSHL	ATTRIBUTES	3974
				08	AE	9F	0016A		PUSHAB	STRING_PTR		
				08	AE	9F	0016D		PUSHAB	CURRENT_LEN		
				B8	A9	9F	00170		PUSHAB	CARRIAGE_FORMAT		
		6A			04	FB	00173		CALLS	#4, SYSSFAO		
		65			6E	A0	00176		ADDW2	CURRENT_LEN, (R5)	3976	
		08			6E	C0	00179		ADDL2	CURRENT_LEN, STRING_PTR+4	3977	
		04			65	3C	0017D		MOVZWL	(R5), STRING_PTR	3978	
04	AE	00000200		8F	04	AE	C3	00181	SUBL3	STRING_PTR, #512, STRING_PTR		
				01	53	91	0018B		CMPB	R3, #1	3980	
					36	13	0018E		BEQL	20\$		
					54	DD	00190		PUSHL	RECORD_SIZE	3987	
				08	AE	9F	00192		PUSHAB	STRING_PTR		
				08	AE	9F	00195		PUSHAB	CURRENT_LEN		
				AC	A9	9F	00198		PUSHAB	REC_SIZE		
		6A			04	FB	0019B		CALLS	#4, SYSSFAO		
		65			6E	A0	0019E		ADDW2	CURRENT_LEN, (R5)	3989	
		08			6E	C0	001A1		ADDL2	CURRENT_LEN, STRING_PTR+4	3990	
		04			65	3C	001A5		MOVZWL	(R5), STRING_PTR	3991	
04	AE	00000200		8F	04	AE	C3	001A9	SUBL3	STRING_PTR, #512, STRING_PTR		
					11	11	001B3		BRB	20\$	3861	
					58	DD	001B5	19\$:	PUSHL	R8	4002	
				08	AE	9F	001B7		PUSHAB	STRING_PTR		
				08	AE	9F	001BA		PUSHAB	CURRENT_LEN		
				F8	A9	9F	001BD		PUSHAB	FORMAT_NEG		
		6A			04	FB	001C0		CALLS	#4, SYSSFAO		
		65			6E	B0	001C3		MOVW	CURRENT_LEN, (R5)	4004	
	0200	8F			65	B1	001C6	20\$:	CMPW	(R5), #512	4008	
					05	1B	001CB		BLEQU	21\$		
		65		0200	8F	B0	001CD		MOVW	#512, (R5)	4011	
					04	001D2	21\$:	RET			4016	

; Routine Size: 467 bytes, Routine Base: CODE + 1775

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_FILE_DESCRIPTION - Create a Sentence Descri

C 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 102
(26)

SE
VO
:

```
3096 4017 1 %sbttl 'GET_FILE_NAME - Get Name of the Current File'
3097 4018 1 ++
3098 4019 1 Functional Description:
3099 4020 1 This routine creates a phrase with the name of the current file.
3100 4021 1
3101 4022 1 Formal Parameters:
3102 4023 1 SCB - Address of the SCB
3103 4024 1 STR_DESC - Desc of String to Return
3104 4025 1 RET_LEN - Return length of Desc.
3105 4026 1
3106 4027 1 Implicit Inputs:
3107 4028 1 none
3108 4029 1
3109 4030 1 Implicit Outputs:
3110 4031 1 none
3111 4032 1
3112 4033 1 Returned Value:
3113 4034 1 none
3114 4035 1
3115 4036 1 Side Effects:
3116 4037 1 none
3117 4038 1 --
3118 4039 1 ROUTINE GET_FILE_NAME (
3119 4040 1 SCB : REF $BLOCK, : SCB
3120 4041 1 EXPECTED_LEN : REF VECTOR[2], : Maximum length allowed
3121 4042 1 STR_DESC : REF VECTOR[2], : Output buffer desc
3122 4043 1 RET_LEN : REF VECTOR[2], : Return length (word)
3123 4044 1 ) : NOVALUE =
3124 4045 2 BEGIN
3125 4046 2 BIND
3126 P 4047 2 SENT_FORMAT = $DESCRIPTOR (
3127 4048 2 'AS');
3128 4049 2
3129 4050 2 LOCAL
3130 4051 2 BUFFER : VECTOR [512,byte],
3131 4052 2 LENG : VECTOR [1],
3132 4053 2 NAME : VECTOR[2];
3133 4054 2
3134 4055 2 NAME[SIZE] = %ALLOCATION(BUFFER); : allocate for routines
3135 4056 2 NAME[ADDR] = BUFFER; : init address
3136 4057 2
3137 P 4058 2 $FAO ( SENT_FORMAT,
3138 P 4059 2 NAME[SIZE], : return length
3139 P 4060 2 NAME, : address of string
3140 P 4061 2 SCB[PSMSQ_FILE_SPECIFICATION], : file name
3141 4062 2 );
3142 4063 2
3143 4064 2 LENG[0] = .EXPECTED_LEN; : must be reference to word for call
3144 4065 2
3145 4066 2 IF .RET_LEN[0] GTR .EXPECTED_LEN THEN
3146 4067 2 Trim the file spec to fit.
3147 4068 2
3148 4069 2 LIB$TRIM_FILESPEC ( NAME, STR_DESC[0], LENG[0],
3149 4070 2 STR_DESC[SIZE])
3150 4071 2 ELSE
3151 4072 2 BEGIN
3152 4073 2 STR_DESC[SIZE] = .NAME[SIZE];
```

; Routine Size: 88 bytes, Routine Base: CODE + 1954


```
3161 4081 1 %sbttl 'INSERT_FILENAME_BANNER - Get Name of the Current File'
3162 4082 1 **
3163 4083 1 Functional Description:
3164 4084 1 This routine creates a banner phrase with the name of the current file.
3165 4085 1 Algorithm:
3166 4086 1 If the Filename, Type, and Version (FTV) fits on one line
3167 4087 1 If only one banner line exists...
3168 4088 1 insert FTV on only ONE line
3169 4089 1
3170 4090 1 If there are Three banner lines available...
3171 4091 1 insert Filename on one, Type on
3172 4092 1 another, and Version on third
3173 4093 1
3174 4094 1 Otherwise...
3175 4095 1 insert Filename on one, Type and
3176 4096 1 Version on the second
3177 4097 1
3178 4098 1 Formal Parameters:
3179 4099 1 SCB - Address of the SCB
3180 4100 1 STR_DESC - Desc of String to Return
3181 4101 1 RET_LEN - Return length of Desc.
3182 4102 1
3183 4103 1 Implicit Inputs:
3184 4104 1 none
3185 4105 1
3186 4106 1 Implicit Outputs:
3187 4107 1 none
3188 4108 1
3189 4109 1 Returned Value:
3190 4110 1 none
3191 4111 1
3192 4112 1 Side Effects:
3193 4113 1 none
3194 4114 1 --
3195 4115 1 ROUTINE INSERT_FILENAME_BANNER (
3196 4116 1 SCB : REF $BLOCK,
3197 4117 1 STR_DESC : REF VECTOR[2],
3198 4118 1 FRAME_PTR : REF PAGE_ARRAY,
3199 4119 1 FRAME_WIDTH , Number of Columns
3200 4120 1 FRAME_LENGTH : Number of Rows
3201 4121 1 ) =
3202 4122 2 BEGIN
3203 4123 2 LITERAL
3204 4124 2 BIG_BANNER = 14,
3205 4125 2 LITTLE_BANNER = 7,
3206 4126 2 SMALL = 2,
3207 4127 2 LARGE = 1;
3208 4128 2
3209 4129 2 LOCAL
3210 4130 2 RET_LEN : VECTOR[1],
3211 4131 2 PAGE_PTR : REF PAGE_ARRAY,
3212 4132 2 SPACING ,
3213 4133 2 CURRENT_PTR ,
3214 4134 2 FTV_LEN : VECTOR[1],
3215 4135 2 BANNER_TYPE ,
3216 4136 2 BANNER_SIZE ,
3217 4137 2 MAX_BAN_CHARS,
3217 4137 2 MAX_ROWS ,
```

```
3218 4138 2 MAX_COLS
3219 4139 3 NAME : VECTOR[2],
3220 4140 3 TYPE : VECTOR[2],
3221 4141 3 VERS : VECTOR[2];
3222 4142 2
3223 4143 2 ! dont even try if there is no frame left
3224 4144 2
3225 4145 2 IF (.FRAME_LENGTH LSS 7)
3226 4146 3 THEN
3227 4147 3 RETURN 0;
3228 4148 2
3229 4149 2 PARSE_FILE_NAME (SCB[PSMSQ_FILE_SPECIFICATION], FSCNS_NAME, NAME);
3230 4150 2 PARSE_FILE_NAME (SCB[PSMSQ_FILE_SPECIFICATION], FSCNS_TYPE, TYPE);
3231 4151 2 PARSE_FILE_NAME (SCB[PSMSQ_FILE_SPECIFICATION], FSCNS_VERSION, VERS);
3232 4152 2
3233 4153 2 FTV_LEN[0] = .NAME[SIZE] + .TYPE[SIZE] + .VERS[SIZE];
3234 4154 2
3235 4155 2 BANNER_TYPE = BIG BANNER;
3236 4156 2 BANNER_SIZE = LARGE;
3237 4157 2 SPACING = SMALL; ! two spaces between banner rows
3238 4158 2
3239 4159 2 MAX_BAN_CHARS = (.FRAME_LENGTH/16) * (.FRAME_WIDTH/12);
3240 4160 2 MAX_ROWS = .BANNER_SIZE * (.FRAME_LENGTH / 16);
3241 4161 2 MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 12);
3242 4162 2
3243 4163 2 IF .MAX_BAN_CHARS LSS .FTV_LEN[0]
3244 4164 3 THEN
3245 4165 4 BEGIN
3246 4166 5 BANNER_SIZE = SMALL;
3247 4167 5 BANNER_TYPE = LITTLE_BANNER;
3248 4168 5 SPACING = LARGE; ! single space banner rows
3249 4169 5 MAX_ROWS = .BANNER_SIZE * (.FRAME_LENGTH / 18);
3250 4170 5 MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 14);
3251 4171 4 END;
3252 4172 2
3253 4173 2
3254 4174 2 ! Do somemore calculations to ensure consistent letter sizing
3255 4175 2 !X! Just to gte this out the door... needs to be optimized later.rb
3256 4176 2
3257 4177 4 IF ( (.NAME[SIZE] GTR .MAX_COLS)
3258 4178 5 OR
3259 4179 5 (.TYPE[SIZE] GTR .MAX_COLS)
3260 4180 5 OR
3261 4181 5 (.VERS[SIZE] GTR .MAX_COLS)
3262 4182 5 OR
3263 4183 5 ( (.MAX_ROWS LSS 3) AND
3264 4184 6 ((.TYPE[SIZE] + .VERS[SIZE]) GTR .MAX_COLS)
3265 4185 5 OR
3266 4186 5 (.FTV_LEN[0] GTR (.MAX_COLS * .MAX_ROWS)) )
3267 4187 4 )
3268 4188 3 THEN
3269 4189 4 BEGIN
3270 4190 5 BANNER_SIZE = SMALL;
3271 4191 5 BANNER_TYPE = LITTLE_BANNER;
3272 4192 5 SPACING = LARGE; ! single space banner rows
3273 4193 5 MAX_ROWS = .BANNER_SIZE * (.FRAME_LENGTH / 18);
3274 4194 5 MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 14);
```

```
3275 4195 2      END;
3276 4196 2
3277 4197 2      ! Attempt to fit the filename, type, and version on one line
3278 4198 2
3279 4199 2      IF (.FTV_LEN[0] LEQ .MAX_COLS)                ! insert on one line
3280 4200 2          AND                                         ! only if little banner
3281 4201 2          (.BANNER_TYPE EQL LITTLE_BANNER)
3282 4202 2      THEN
3283 4203 2          BEGIN
3284 4204 2              CURRENT_PTR = .STR_DESC[ADDR];
3285 4205 2              CURRENT_PTR = CH$MOVE(.NAME[SIZE], .NAME[ADDR], .CURRENT_PTR);
3286 4206 2              CURRENT_PTR = CH$MOVE(.TYPE[SIZE], .TYPE[ADDR], .CURRENT_PTR);
3287 4207 2              CURRENT_PTR = CH$MOVE(.VERS[SIZE], .VERS[ADDR], .CURRENT_PTR);
3288 4208 2              STR_DESC[SIZE] = .FTV_LEN[0];
3289 4209 2
3290 4210 2              INSERT_NAME_BANNER (.SCB,
3291 4211 2                  STR_DESC[SIZE],                ! job name desc
3292 4212 2                  FRAME_PTR[0,0,.SCB[PSMSL PAGE_WIDTH]],
3293 4213 2                  ! ref to frame
3294 4214 2                  .FRAME_WIDTH,                ! max width Bann
3295 4215 2                  .BANNER_TYPE,                ! frame length
3296 4216 2                  .BANNER_TYPE);                ! max hght Bann str
3297 4217 2
3298 4218 2          RETURN .BANNER_TYPE;                ! return how much space used
3299 4219 2      END
3300 4220 2      ELSE
3301 4221 2          BEGIN
3302 4222 2              ! Move filename with truncated banners when not enough space
3303 4223 2
3304 4224 2              IF ( (.MAX_ROWS LEQ 1) )
3305 4225 2
3306 4226 2              !X! Comment this out .... causes too many filenames to be printed on
3307 4227 2              !X! a single line when two lines would be more appropriate.
3308 4228 2
3309 4229 2              OR
3310 4230 2              (.NAME[SIZE] GTR .MAX_COLS)
3311 4231 2              OR
3312 4232 2              (.TYPE[SIZE] GTR .MAX_COLS)
3313 4233 2              OR
3314 4234 2              (.VERS[SIZE] GTR .MAX_COLS)
3315 4235 2              OR
3316 4236 2              ( (.MAX_ROWS LSS 3) AND
3317 4237 2              ((.TYPE[SIZE]+.VERS[SIZE]) GTR .MAX_COLS)
3318 4238 2              OR
3319 4239 2              (.FTV_LEN[0] GTR (.MAX_COLS * .MAX_ROWS)) )
3320 4240 2          )
3321 4241 2      THEN
3322 4242 2          BEGIN
3323 4243 2              CURRENT_PTR = .STR_DESC[ADDR];
3324 4244 2              CURRENT_PTR = CH$MOVE(.NAME[SIZE], .NAME[ADDR], .CURRENT_PTR);
3325 4245 2              CURRENT_PTR = CH$MOVE(.TYPE[SIZE], .TYPE[ADDR], .CURRENT_PTR);
3326 4246 2              CURRENT_PTR = CH$MOVE(.VERS[SIZE], .VERS[ADDR], .CURRENT_PTR);
3327 4247 2
3328 4248 2          !X! This is a cludge to get this out the door. I will declare a valid
3329 4249 2          ! descriptor in the future and use LIB$TRIM_FILESPEC.
3330 4250 2
3331 4251 2          STR_DESC[SIZE] = .FTV_LEN[0];
```

```
3332 4252 4
3333 4253 4 IF .NAME[0] GEQ .MAX_COLS
3334 4254 4 THEN
3335 4255 5 (STR_DESC[SIZE] = .MAX_COLS)
3336 4256 5 ELSE
3337 4257 5 (IF .NAME[0]+.TYPE[0] GTR .MAX_COLS
3338 4258 5 THEN
3339 4259 5 STR_DESC[SIZE] = .NAME[0]
3340 4260 5 ELSE
3341 4261 6 IF ((.NAME[0]+.TYPE[0]+.VERS[0]) GTR .MAX_COLS)
3342 4262 5 THEN
3343 4263 4 STR_DESC[SIZE] = .NAME[0]+.TYPE[0]);
3344 4264 4
3345 4265 4 IF .FTV_LEN[0] GTR (.MAX_ROWS * .MAX_COLS)
3346 4266 4 THEN
3347 4267 4 ! Trim the file spec to fit.
3348 4268 4
3349 4269 4 LIB$TRIM_FILESPEC ( STR_DESC, STR_DESC[0],
3350 4270 4 .MAX_COLS,
3351 4271 4 STR_DESC[SIZE]);
3352 4272 4
3353 4273 4
3354 4274 4 RET_LEN[0] = INSERT_NAME_BANNER (
3355 4275 4 .SCB,
3356 4276 4 STR_DESC[SIZE], ! job name desc
3357 4277 4 FRAME_PTR[0,0,.SCB[PSM$PAGE_WIDTH]],
3358 4278 4 ! ref to frame
3359 4279 4 .FRAME_WIDTH, ! max width Bann
3360 4280 4 .BANNER_TYPE, ! frame length
3361 4281 4 .BANNER_TYPE); ! max hght Bann str
3362 4282 4
3363 4283 4 RETURN .RET_LEN[0]; ! return how much space used
3364 4284 4 END
3365 4285 3 ELSE ! Should be able to insert it... Make it pretty
3366 4286 4 BEGIN
3367 4287 4 IF .MAX_ROWS GEQ 3
3368 4288 4 THEN
3369 4289 5 BEGIN
3370 4290 5
3371 4291 5 PAGE_PTR = FRAME_PTR[0,0,.SCB[PSM$PAGE_WIDTH]];
3372 4292 5 INSERT_NAME_BANNER (
3373 4293 5 .SCB,
3374 4294 5 NAME[SIZE], ! file name
3375 4295 5 PAGE_PTR[0,0,.SCB[PSM$PAGE_WIDTH]],
3376 4296 5 ! ref to frame
3377 4297 5 .FRAME_WIDTH, ! max width Bann
3378 4298 5 .BANNER_TYPE, ! frame length
3379 4299 5 .BANNER_TYPE); ! max hght Bann str
3380 4300 5
3381 4301 5 PAGE_PTR = PAGE_PTR[0, (.BANNER_TYPE+.SPACING),
3382 4302 5 .SCB[PSM$PAGE_WIDTH]];
3383 4303 5 INSERT_NAME_BANNER (
3384 4304 5 .SCB,
3385 4305 5 TYPE[SIZE], ! file type
3386 4306 5 PAGE_PTR[0,0,.SCB[PSM$PAGE_WIDTH]],
3387 4307 5 ! ref to frame
3388 4308 5 .FRAME_WIDTH, ! max width Bann
```



```
.. 3389      4309      5      .BANNER_TYPE,      ! frame length
.. 3390      4310      5      .BANNER_TYPE);      ! max hght Bann str
.. 3391      4311      5
.. 3392      4312      5      PAGE_PTR = PAGE_PTR[0, (.BANNER_TYPE+.SPACING),
.. 3393      4313      5      .SCB[PSM$L_PAGE_WIDTH]];
.. 3394      4314      5      INSERT_NAME_BANNER (
.. 3395      4315      5      .SCB,
.. 3396      4316      5      VERS[SIZE],      ! file version number
.. 3397      4317      5      PAGE_PTR[0,0,.SCB[PSM$L_PAGE_WIDTH]],
.. 3398      4318      5      .FRAME_WIDTH,      ! ref to frame
.. 3399      4319      5      .BANNER_TYPE,      ! max width Bann
.. 3400      4320      5      .BANNER_TYPE);      ! frame length
.. 3401      4321      5      ! max hght Bann str
.. 3402      4322      5
.. 3403      4323      5      RETURN (3 * (.BANNER_TYPE + .SPACING));      ! return count of used
.. 3404      4324      5      ! space
.. 3405      4325      5      END
.. 3406      4326      5      ELSE
.. 3407      4327      5      BEGIN
.. 3408      4328      5      CURRENT_PTR = .STR_DESC[ADDR];
.. 3409      4329      5      CURRENT_PTR = CH$MOVE(.TYPE[SIZE], .TYPE[ADDR], .CURRENT_PTR);
.. 3410      4330      5      CURRENT_PTR = CH$MOVE(.VERS[SIZE], .VERS[ADDR], .CURRENT_PTR);
.. 3411      4331      5      STR_DESC[SIZE] = .TYPE[SIZE] + .VERS[SIZE];
.. 3412      4332      5
.. 3413      4333      5      PAGE_PTR = FRAME_PTR[0,0,.SCB[PSM$L_PAGE_WIDTH]];
.. 3414      4334      5      INSERT_NAME_BANNER (
.. 3415      4335      5      .SCB,
.. 3416      4336      5      NAME[SIZE],      ! file name
.. 3417      4337      5      PAGE_PTR[0,0,.SCB[PSM$L_PAGE_WIDTH]],
.. 3418      4338      5      .FRAME_WIDTH,      ! ref to frame
.. 3419      4339      5      .BANNER_TYPE,      ! max width Bann
.. 3420      4340      5      .BANNER_TYPE);      ! frame length
.. 3421      4341      5      ! max hght Bann str
.. 3422      4342      5
.. 3423      4343      5      PAGE_PTR = PAGE_PTR[0, (.BANNER_TYPE+.SPACING),
.. 3424      4344      5      .SCB[PSM$L_PAGE_WIDTH]];
.. 3425      4345      5      INSERT_NAME_BANNER (
.. 3426      4346      5      .SCB,
.. 3427      4347      5      STR_DESC[SIZE],      ! file and version type
.. 3428      4348      5      PAGE_PTR[0,0,.SCB[PSM$L_PAGE_WIDTH]],
.. 3429      4349      5      .FRAME_WIDTH,      ! ref to frame
.. 3430      4350      5      .BANNER_TYPE,      ! max width Bann
.. 3431      4351      5      .BANNER_TYPE);      ! frame length
.. 3432      4352      5      ! max hght Bann str
.. 3433      4353      5
.. 3434      4354      5      RETURN (2 * (.BANNER_TYPE + .SPACING));      ! return count of used
.. 3435      4355      5      ! space
.. 3436      4356      5      END;
.. 3437      4357      5      END;
.. 3438      4358      5      END;
.. 3439      4359      1      END;
```

OFFC 00000 INSERT_FILENAME_BANNER:

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_FILENAME_BANNER - Get Name of the Curren

K 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 110
(28)

		5F		20	C2	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	4114
		07		14	AC	D1 00005	SUBL2	#32, SP	
				03	18	00009	CMPL	FRAME_LENGTH, #7	4145
				0211	31	0000B	BGEQ	1\$	
			18	AE	9F	0000E	BRW	12\$	
				06	DD	00011	PUSHAB	NAME	4149
50	04	AC	00000098	8F	C1	00013	PUSHL	#6	
		52		60	9E	0001C	ADDL3	#152, SCB, R0	
				52	DD	0001F	MOVAB	(R0), R2	
	EC5F	CF		03	FB	00021	PUSHL	R2	
			10	AE	9F	00026	CALLS	#3, PARSE_FILE_NAME	
				07	DD	00029	PUSHAB	TYPE	4150
				52	DD	0002B	PUSHL	#7	
	EC53	CF		03	FB	0002D	PUSHL	R2	
			08	AE	9F	00032	CALLS	#3, PARSE_FILE_NAME	
				08	DD	00035	PUSHAB	VERS	4151
				52	DD	00037	PUSHL	#8	
	EC47	CF		03	FB	00039	PUSHL	R2	
		5B		18	AE	D0 0003E	CALLS	#3, PARSE_FILE_NAME	
	04	AE		10	BE4B	9E 00042	MOVL	NAME, R11	4153
6E	04	AE		08	AE	C1 00048	MOVAB	@TYPE[R11], 4(SP)	
		57			0E	D0 0004E	ADDL3	VERS, 4(SP), FTV_LEN	
		50			01	D0 00051	MOVL	#14, BANNER_TYPE	4155
		54			02	D0 00054	MOVL	#1, BANNER_SIZE	4156
52	14	AC		10	C7	00057	MOVL	#2, SPACING	4157
		5A		10	AC	D0 0005C	DIVL3	#16, FRAME_LENGTH, R2	4159
51		5A			0C	C7 00060	MOVL	FRAME_WIDTH, R10	
53		52			51	C5 00064	DIVL3	#12, R10, R1	
59		50			52	C5 00068	MULL3	R1, R2, MAX_BAN_CHARS	
56		50			51	C5 0006C	MULL3	R2, BANNER_SIZE, MAX_ROWS	4160
		6E			53	D1 00070	MULL3	#1, BANNER_SIZE, MAX_COLS	4161
					1A	18 00073	CMPL	MAX_BAN_CHARS, FTV_LEN	4163
		50			02	D0 00075	BGEQ	2\$	
		57			07	D0 00078	MOVL	#2, BANNER_SIZE	4166
		54			01	D0 0007B	MOVL	#7, BANNER_TYPE	4167
51	14	AC			12	C7 0007E	MOVL	#1, SPACING	4168
59		51			50	C5 00083	DIVL3	#18, FRAME_LENGTH, R1	4169
51		5A			0E	C7 00087	MULL3	BANNER_SIZE, R1, MAX_ROWS	
56		51			50	C5 0008B	DIVL3	#14, R10, R1	4170
		56			5B	D1 0008F	MULL3	BANNER_SIZE, R1, MAX_COLS	
					25	14 00092	CMPL	R11, MAX_COLS	4177
		56		10	AE	D1 00094	BGTR	4\$	
					1F	14 00098	CMPL	TYPE, MAX_COLS	4179
		56		08	AE	D1 0009A	BGTR	4\$	
					19	14 0009E	CMPL	VERS, MAX_COLS	4181
		03			59	D1 000A0	BGTR	4\$	
					0B	18 000A3	CMPL	MAX_ROWS, #3	4183
51	10	AE		08	AE	C1 000A5	BGEQ	3\$	
		56			51	D1 000AB	ADDL3	VERS, TYPE, R1	4184
					09	14 000AE	CMPL	R1, MAX_COLS	
51		56			59	C5 000B0	BGTR	4\$	
		51			6E	D1 000B4	MULL3	MAX_ROWS, MAX_COLS, R1	4186
					1A	15 000B7	CMPL	FTV_LEN, R1	
		50			02	D0 000B9	BLEQ	5\$	
		57			07	D0 000BC	MOVL	#2, BANNER_SIZE	4190
		54			01	D0 000BF	MOVL	#7, BANNER_TYPE	4191
							MOVL	#1, SPACING	4192

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_FILENAME_BANNER - Get Name of the

L 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 111
(28)

51	14	AC	12	C7	000C2	DIVL3	#18, FRAME_LENGTH, R1	4193	
59		51	50	C5	000C7	MULL3	BANNER_SIZE, R1, MAX_ROWS		
51		5A	0E	C7	000CB	DIVL3	#14, RTO, R1	4194	
56		51	50	C5	000CF	MULL3	BANNER_SIZE, R1, MAX_COLS		
		56	6E	D1	000D3	CMPL	FTV_LEN, MAX_COLS	4199	
			38	14	000D6	BGTR	6\$		
		07	57	D1	000D8	CMPL	BANNER_TYPE, #7	4201	
			33	12	000DB	BNEQ	6\$		
		58	AC	D0	000DD	MOVL	STR_DESC, R8	4204	
		53	04	A8	D0	000E1	MOVL	4(R8), CURRENT_PTR	
63	1C	BE	5B	28	000E5	MOVC3	R11, @NAME+4, (CURRENT_PTR)	4205	
63	14	BE	AE	28	000EA	MOVC3	TYPE, @TYPE+4, (CURRENT_PTR)	4206	
63	0C	BE	AE	28	000F0	MOVC3	VERS, @VERS+4, (CURRENT_PTR)	4207	
		68	6E	D0	000F6	MOVL	FTV_LEN, (R8)	4208	
			57	DD	000F9	PUSHL	BANNER_TYPE	4216	
			57	DD	000FB	PUSHL	BANNER_TYPE	4215	
			5A	DD	000FD	PUSHL	R10	4214	
			0C	AC	DD	000FF	PUSHL	FRAME_PTR	4212
			58	DD	00102	PUSHL	R8		
			04	AC	DD	00104	PUSHL	SCB	
	0000V	CF	06	FB	00107	CALLS	#6, INSERT_NAME_BANNER		
		50	57	D0	0010C	MOVL	BANNER_TYPE, R0	4221	
				04	0010F	RET			
		01	59	D1	00110	CMPL	MAX_ROWS, #1	4224	
			5A	14	00113	BGTR	10\$		
		58	AC	D0	00115	MOVL	STR_DESC, R8	4243	
		53	04	A8	D0	00119	MOVL	4(R8), CURRENT_PTR	
63	1C	BE	5B	28	0011D	MOVC3	R11, @NAME+4, (CURRENT_PTR)	4244	
63	14	BE	AE	28	00122	MOVC3	TYPE, @TYPE+4, (CURRENT_PTR)	4245	
63	0C	BE	AE	28	00128	MOVC3	VERS, @VERS+4, (CURRENT_PTR)	4246	
		68	6E	D0	0012E	MOVL	FTV_LEN, (R8)	4251	
		56	5B	D1	00131	CMPL	R11, MAX_COLS	4253	
			05	19	00134	BLSS	7\$		
		68	56	D0	00136	MOVL	MAX_COLS, (R8)	4255	
			20	11	00139	BRB	9\$		
		56	04	AE	D1	0013B	CMPL	4(SP), MAX_COLS	4257
			05	15	0013F	BLEQ	8\$		
		68	5B	D0	00141	MOVL	R11, (R8)	4259	
			15	11	00144	BRB	9\$		
		50	10	AE	9E	00146	MOVAB	TYPE, R0	4261
		50	08	AE	40	9E	0014A	MOVAB	VERS[R0], R0
		50		5B	C0	0014F	ADDL2	R11, R0	
		56		50	D1	00152	CMPL	R0, MAX_COLS	
			04	15	00155	BLEQ	9\$		
		68	04	AE	D0	00157	MOVL	4(SP), (R8)	4263
			57	DD	0015B	PUSHL	BANNER_TYPE	4281	
			57	DD	0015D	PUSHL	BANNER_TYPE	4280	
			5A	DD	0015F	PUSHL	R10	4279	
			0C	AC	DD	00161	PUSHL	FRAME_PTR	4277
			58	DD	00164	PUSHL	R8		
			04	AC	DD	00166	PUSHL	SCB	
	0000V	CF	06	FB	00169	CALLS	#6, INSERT_NAME_BANNER		
				04	0016E	RET		4286	
		56	0C	AC	D0	0016F	MOVL	FRAME_PTR, PAGE_PTR	4291
58		57		54	C1	00173	ADDL3	SPACING, BANNER_TYPE, R8	4301
50	04	AC	00000200	8F	C1	00177	ADDL3	#512, SCB, R0	4302
		5B		60	9E	00180	MOVAB	(R0), R11	

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_FILENAME_BANNER - Get Name of the

M 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 112
(28)

		03		59	D1	00183		CMPL	MAX_ROWS, #3		4287
				4C	19	00186		BLSS	11\$		
				57	DD	00188		PUSHL	BANNER_TYPE		4299
				57	DD	0018A		PUSHL	BANNER_TYPE		4298
		0440		8F	BB	0018C		PUSHR	#*M<R6,R10>		4295
		28		AE	9F	00190		PUSHAB	NAME		4294
		04		AC	DD	00193		PUSHL	SCB		4295
50	0000V	CF		06	FB	00196		CALLS	#6, INSERT_NAME_BANNER		
		58		6B	C5	0019B		MULL3	(R11), R8, R0		4302
		56		50	C0	0019F		ADDL2	R0, PAGE_PTR		
				57	DD	001A2		PUSHL	BANNER_TYPE		4310
				57	DD	001A4		PUSHL	BANNER_TYPE		4309
				8F	BB	001A6		PUSHR	#*M<R6,R10>		4306
		0440		AE	9F	001AA		PUSHAB	TYPE		4305
		20		AC	DD	001AD		PUSHL	SCB		4306
50	0000V	CF		06	FB	001B0		CALLS	#6, INSERT_NAME_BANNER		
		58		6B	C5	001B5		MULL3	(R11), R8, R0		4313
		56		50	C0	001B9		ADDL2	R0, PAGE_PTR		
				57	DD	001BC		PUSHL	BANNER_TYPE		4321
				57	DD	001BE		PUSHL	BANNER_TYPE		4320
				8F	BB	001C0		PUSHR	#*M<R6,R10>		4317
		0440		AE	9F	001C4		PUSHAB	VERS		4316
		18		AC	DD	001C7		PUSHL	SCB		4317
50	0000V	CF		06	FB	001CA		CALLS	#6, INSERT_NAME_BANNER		
		58		03	C5	001CF		MULL3	#3, R8, R0		4323
				04	001D3			RET			4327
		59		08	AC	001D4	11\$:	MOVL	STR_DESC, R9		4328
		53		04	A9	001D8		MOVL	4(R9), CURRENT_PTR		
63	14	BE		10	AE	001DC		MOVC3	TYPE, @TYPE+4, (CURRENT_PTR)		4329
63	0C	BE		08	AE	001E2		MOVC3	VERS, @VERS+4, (CURRENT_PTR)		4330
69	10	AE		08	AE	001E8		ADDL3	VERS, TYPE, (R9)		4331
				57	DD	001EE		PUSHL	BANNER_TYPE		4341
				57	DD	001F0		PUSHL	BANNER_TYPE		4340
				8F	BB	001F2		PUSHR	#*M<R6,R10>		4337
		0440		AE	9F	001F6		PUSHAB	NAME		4336
		28		AC	DD	001F9		PUSHL	SCB		4337
50	0000V	CF		06	FB	001FC		CALLS	#6, INSERT_NAME_BANNER		
		58		6B	C5	00201		MULL3	(R11), R8, R0		4344
		56		50	C0	00205		ADDL2	R0, PAGE_PTR		
				57	DD	00208		PUSHL	BANNER_TYPE		4352
				57	DD	0020A		PUSHL	BANNER_TYPE		4351
				8F	BB	0020C		PUSHR	#*M<R6,R10>		4348
		0440		59	DD	00210		PUSHL	R9		
		04		AC	DD	00212		PUSHL	SCB		
50	0000V	CF		06	FB	00215		CALLS	#6, INSERT_NAME_BANNER		
		58		01	78	0021A		ASHL	#1, R8, R0		4354
				04	0021E			RET			4221
				50	D4	0021F	12\$:	CLRL	R0		4359
				04	00221			RET			

; Routine Size: 546 bytes. Routine Base: CODE + 19AC


```
3441 4360 1 %sbttl 'INSERT_JOBNUMBER_BANNER - Get Job Number of the current Job'
3442 4361 1 ++
3443 4362 1 Functional Description:
3444 4363 1 This routine creates a banner phrase with the Job Number
3445 4364 1
3446 4365 1 Formal Parameters:
3447 4366 1 SCB - Address of the SCB
3448 4367 1 STR_DESC - Desc of String to Return
3449 4368 1 RET_LEN - Return length of Desc.
3450 4369 1
3451 4370 1 Implicit Inputs:
3452 4371 1 none
3453 4372 1
3454 4373 1 Implicit Outputs:
3455 4374 1 none
3456 4375 1
3457 4376 1 Returned Value:
3458 4377 1 none
3459 4378 1
3460 4379 1 Side Effects:
3461 4380 1 none
3462 4381 1 --
3463 4382 1 ROUTINE INSERT_JOBNUMBER_BANNER (
3464 4383 1 SCB : REF $BBLOCK,
3465 4384 1 STR_DESC : REF VECTOR[2],
3466 4385 1 FRAME_PTR : REF PAGE_ARRAY,
3467 4386 1 FRAME_WIDTH , Number of Columns
3468 4387 1 FRAME_LENGTH : Number of Rows
3469 4388 1 ) =
3470 4389 2 BEGIN
3471 4390 2 BIND
3472 P 4391 2 SENT_FORMAT = $DESCRIPTOR (
3473 P 4392 2 'JOB ',
3474 4393 2 '!UL');
3475 4394 2
3476 P 4395 2 NUM_FORMAT = $DESCRIPTOR (
3477 4396 2 '!UL');
3478 4397 2
3479 4398 2 LITERAL
3480 4399 2 LITTLE_BANNER = 7,
3481 4400 2 SMALL = 2,
3482 4401 2 LARGE = 1;
3483 4402 2
3484 4403 2 LOCAL
3485 4404 2 RET_LEN : VECTOR[1],
3486 4405 2 PAGE_PTR : REF PAGE_ARRAY,
3487 4406 2 CURRENT_PTR ,
3488 4407 2 JOB_LEN : INITIAL (0),
3489 4408 2 BANNER_TYPE ,
3490 4409 2 SPACING ,
3491 4410 2 BANNER_SIZE ,
3492 4411 2 MAX_BAN_CHARS,
3493 4412 2 BUFFER : VECTOR[10,byte],
3494 4413 2 MAX_COLS ,
3495 4414 2 NUMBER : VECTOR[2];
3496 4415 2
3497 4416 2 NUMBER[SIZE] = %ALLOCATION(BUFFER);
```

```

3498      4417      2  NUMBER[ADDR] = BUFFER;
3499      4418      2
3500      4419      2  ! dont even try if there is no frame left
3501      4420      2  !
3502      4421      2  IF (.FRAME_LENGTH LSS 7)
3503      4422      2  THEN
3504      4423      2      RETURN 0;
3505      4424      2
3506      4425      2  MAX_BAN_CHARS = (.FRAME_LENGTH/16) * (.FRAME_WIDTH/12);
3507      4426      2
3508      4427      2  BANNER_SIZE = SMALL;
3509      4428      2  BANNER_TYPE = LITTLE_BANNER;
3510      4429      2  SPACING = LARGE;
3511      4430      2                                     ! single space banner rows
3512      4431      2  MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 12);
3513      4432      2
3514      4433      2
3515      4434      2  $FAO (  NUM_FORMAT,
3516      4435      2      JOB_LEN,                                     ! str[size] > fetched namelen
3517      4436      2      NUMBER[0],
3518      4437      2      .SCB[PSM$$_ENTRY_NUMBER]
3519      4438      2      );
3520      4439      2
3521      4440      2  IF (.JOB_LEN+4) LEQ .MAX_COLS
3522      4441      2      THEN
3523      4442      2      BEGIN
3524      4443      2          $FAO (  SENT_FORMAT,
3525      4444      2              JOB_LEN,
3526      4445      2              STR_DESC[0],
3527      4446      2              .SCB[PSM$$_ENTRY_NUMBER]
3528      4447      2              );
3529      4448      2      STR_DESC[SIZE] = .JOB_LEN;
3530      4449      2      END
3531      4450      2      ELSE
3532      4451      2      BEGIN
3533      4452      2          IF .JOB_LEN LEQ .MAX_COLS
3534      4453      2          THEN
3535      4454      2          BEGIN
3536      4455      2              CURRENT_PTR = .STR_DESC[ADDR];
3537      4456      2              CURRENT_PTR = CHSMOVE(.JOB_LEN, .NUMBER[ADDR], .CURRENT_PTR);
3538      4457      2              STR_DESC[SIZE] = .JOB_LEN;
3539      4458      2          END
3540      4459      2          ELSE
3541      4460      2              RETURN 0;
3542      4461      2          END;
3543      4462      2
3544      4463      2  INSERT_NAME_BANNER (
3545      4464      2      .SCB,
3546      4465      2      STR_DESC[SIZE],
3547      4466      2      FRAME_PTR[0,0,.SCB[PSM$$_PAGE_WIDTH]],
3548      4467      2      .FRAME_WIDTH,
3549      4468      2      .BANNER_TYPE,
3550      4469      2      .BANNER_TYPE);
3551      4470      2
3552      4471      2
3553      4472      2  RETURN .BANNER_TYPE;
3554      4473      2

```

SEPARATE
V04-001

; 3555

Print Symbiont -- separation routines
INSERT_JOBNUMBER_BANNER - Get Job Number of the

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 115
(29)

4474 1 END;

```
20 42 4F 4A 01BCE P.ACK: .ASCII \JOB \
    4C 55 21 01BD2 .ASCII \!UL\
          01BD5 .BLKB 3
          00000007 01BD8 P.ACJ: .LONG 7
          00000000 01BDC .ADDRESS P.ACK
    4C 55 21 01BE0 P.ACM: .ASCII \!UL\
          01BE3 .BLKB 1
          00000003 01BE4 P.ACL: .LONG 3
          00000000 01BE8 .ADDRESS P.ACM
```

SENT_FORMAT= P.ACJ
NUM_FORMAT= P.ACL

```
03FC 00000 INSERT_JOBNUMBER_BANNER:
59 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9 4382
5E          14 C2 00009 MOVAB SYSSFA0, R9
          7E D4 0000C SUBL2 #20, SP
04 AE          0A D0 0000E CLRL JOB_LEN 4389
08 AE          0C AE 9E 00012 MOVL #10, NUMBER 4416
07          14 AC D1 00017 MOVAB BUFFER, NUMBER+4 4417
          76 19 0001B CMPL FRAME_LENGTH, #7 4421
51          10 AC          10 C7 0001D BLSS 3$
50          10 AC          0C C7 00022 DIVL3 #16, FRAME_LENGTH, R1 4425
          51          50 C4 00027 DIVL3 #12, FRAME_WIDTH, R0
          51          02 D0 0002A MULL2 R0, MAX_BAN_CHARS
          58          07 D0 0002D MOVL #2, BANNER_SIZE 4427
          52          01 D0 00030 MOVL #7, BANNER_TYPE 4428
          51          50 C5 00033 MOVL #1, SPACING 4429
52          57          04 AC D0 00037 MULL3 R0, BANNER_SIZE, MAX_COLS 4431
          58          08 A7 DD 0003B MOVL SCB, R7 4438
          08          08 AE 9F 0003E PUSHL 88(R7)
          08          08 AE 9F 00041 PUSHAB NUMBER
          B1          AF 9F 00044 PUSHAB JOB_LEN
          69          04 FB 00047 PUSHAB NUM_FORMAT
50          6E          04 C1 0004A CALLS #4, SYSSFA0
          52          50 D1 0004E ADDL3 #4, JOB_LEN, R0 4440
          15 14 00051 CMPL R0, MAX_COLS
          58          A7 DD 00053 BGTR 1$
          08          AC DD 00056 PUSHL 88(R7) 4447
          08          AE 9F 00059 PUSHL STR_DESC
          8D          AF 9F 0005C PUSHAB JOB_LEN
          69          04 FB 0005F PUSHAB SENT_FORMAT
08          BC          6E D0 00062 CALLS #4, SYSSFA0
          52          15 11 00066 MOVL JOB_LEN, @STR_DESC 4448
          56          08 AC D0 0006D BRB 2$ 4440
63          53          04 A6 D0 00071 CMPL JOB_LEN, MAX_COLS 4452
          BE          6E D0 00075 BGTR 3$
          66          58 DD 0007D MOVL STR_DESC, R6 4455
          2$:          4(R6), CURRENT_PTR
          MOVL JOB_LEN, @NUMBER+4, (CURRENT_PTR) 4456
          MOVL JOB_LEN, (R6) 4457
          PUSHL BANNER_TYPE 4470
```

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_JOBNUMBER_BANNER - Get Job Number of the

D 6
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 116
(29)

		58	DD	0007F	PUSHL	BANNER TYPE	
7E	0C	AC	7D	00081	MOVQ	FRAME_PTR, -(SP)	4469
	08	AC	DD	00085	PUSHL	STR_DESC	4466
		57	DD	00088	PUSHL	R7	
0000V	CF	06	FB	0008A	CALLS	#6, INSERT_NAME_BANNER	
	50	58	D0	0008F	MOVL	BANNER_TYPE, R0	4472
			04	00092	RET		
		50	D4	00093	CLRL	R0	4474
			04	00095	RET		

; Routine Size: 150 bytes, Routine Base: CODE + 1BEC

SE
VO


```
3557 4475 1 %sbtll 'GET_JOB_NAME - Get Name of the Current Job'
3558 4476 1 ++
3559 4477 1 Functional Description:
3560 4478 1 This routine creates a phrase with the name of the current job.
3561 4479 1
3562 4480 1 Formal Parameters:
3563 4481 1 SCB - Address of the SCB
3564 4482 1 STR_DESC - Desc of String to Return
3565 4483 1 RET_LEN - Return length of Desc.
3566 4484 1
3567 4485 1 Implicit Inputs:
3568 4486 1 none
3569 4487 1
3570 4488 1 Implicit Outputs:
3571 4489 1 none
3572 4490 1
3573 4491 1 Returned Value:
3574 4492 1 none
3575 4493 1
3576 4494 1 Side Effects:
3577 4495 1 none
3578 4496 1 --
3579 4497 1 ROUTINE GET_JOB_NAME (
3580 4498 1 SCB : REF $BBLOCK, : SCB
3581 4499 1 STR_DESC : REF VECTOR[2], : Output buffer desc
3582 4500 1 RET_LEN : REF VECTOR [,WORD] : Return length (word)
3583 4501 1 ) : NOVALUE =
3584 4502 2 BEGIN
3585 4503 2 BIND
3586 P 4504 2 SENT80_FORMAT = $DESCRIPTOR (
3587 4505 2 '!AS');
3588 4506 2
3589 P 4507 2 $FAO ( SENT80_FORMAT,
3590 P 4508 2 RET_LEN[0], ! str[size] > fetched namelen
3591 P 4509 2 STR_DESC[0],
3592 P 4510 2 SCB[PSM$Q_JOB_NAME], ! job name
3593 4511 2 );
3594 4512 2
3595 4513 2 RETURN SS$_NORMAL;
3596 4514 1 END;
```

```
53 41 21 01C82 P.ACO: .ASCII \!AS\
01C85 .BLKB 3
00000003 01C88 P.ACN: .LONG 3
00000000 01C8C .ADDRESS P.ACO
```

SENT80_FORMAT= P.ACN

```
0000 00000 GET_JOB_NAME:
7E 04 AC 000000A8 8F C1 00002 .WORD Save nothing
08 AC DD 0000B ADDL3 #168, SCB, -(SP)
0C AC DD 0000E PUSHL STR_DESC
E4 AF 9F 00011 PUSHL RET_LEN
PUSHAB SENT80_FORMAT
```

4497
4511

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_JOB_NAME - Get Name of the Current Job

F 6
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 118
(30)

00000000G 00

04 FB 00014
04 0001B

CALLS #4, SYSSFAO
RET

: 4514

; Routine Size: 28 bytes, Routine Base: CODE + 1C90

SE
VO

```
3598 4515 1 %sbtcl 'GET_EOJ - Get the Phrase End of Job'
3599 4516 1 ++
3600 4517 1 Functional Description:
3601 4518 1 This routine creates a phrase with "EOJ" or "END OF JOB".
3602 4519 1
3603 4520 1 Formal Parameters:
3604 4521 1 SCB - Address of the SCB
3605 4522 1 STR_DESC - Desc of String to Return
3606 4523 1 RET_LEN - Return length of Desc.
3607 4524 1
3608 4525 1 Implicit Inputs:
3609 4526 1 none
3610 4527 1
3611 4528 1 Implicit Outputs:
3612 4529 1 none
3613 4530 1
3614 4531 1 Returned Value:
3615 4532 1 none
3616 4533 1
3617 4534 1 Side Effects:
3618 4535 1 none
3619 4536 1 --
3620 4537 1 ROUTINE GET_EOJ (
3621 4538 1 SCB : REF $BLOCK, : SCB
3622 4539 1 STR_DESC : REF VECTOR[2], : Output buffer desc
3623 4540 1 RET_LEN : REF VECTOR [WORD] : Return length (word)
3624 4541 1 ) : NOVALUE =
3625 4542 2 BEGIN
3626 4543 2 BIND
3627 4544 2 SENT132 FORMAT = $DESCRIPTOR (
3628 4545 2 'END OF JOB'), ! -
3629 4546 2
3630 4547 2 SENT80 FORMAT = $DESCRIPTOR (
3631 4548 2 'EOJ');
3632 4549 2
3633 4550 2 $FAO ( SENT132 FORMAT,
3634 4551 2 RET_LEN[0], ! return length
3635 4552 2 STR_DESC[0], ! address of string
3636 4553 2 );
3637 4554 2
3638 4555 2 ! Is it short enough to allow the words "End of Job" to be printed ?
3639 4556 2
3640 4557 2 IF ((12 * .RET_LEN[0]) GTR .SCB[PSM$L_PAGE_WIDTH])
3641 4558 2 THEN
3642 4559 2 $FAO ( SENT80 FORMAT,
3643 4560 2 RET_LEN[0], ! str[size] > fetched namelen
3644 4561 2 STR_DESC[0],
3645 4562 2 );
3646 4563 2
3647 4564 2 RETURN SSS_NORMAL;
3648 4565 1 END;
```

```
42 4F 4A 20 46 4F 20 44 4E 45 01CAC P.ACQ: .ASCII \END OF JOB\
01CB6 .BLKB 2
0000000A 01CB8 P.ACP: .LONG 10
```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_EOJ - Get the Phrase End of Job

H 6
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 120
(31)

SE
VO

```
00000000' 01CBC .ADDRESS P.ACP
4A 4F 45 01CC0 P.ACS: .ASCII \EOJ\
01CC3 .BLK8 1
00000003' 01CC4 P.ACR: .LONG 3
00000000' 01CC8 .ADDRESS P.ACS
```

```
SENT132_FORMAT= P.ACP
SENT80_FORMAT= P.ACR
```

```

      0004 00000 GET_EOJ: .WORD      Save R2
52 00000000G 00 9E 00002 MOVAB     SYSSFAO, R2
      08 AC DD 00009 PUSHL     STR_DESC
      0C AC DD 0000C PUSHL     RET_LEN
      DA AF 9F 0000F PUSHAB    SENT132_FORMAT
      62 03 FB 00012 CALLS     #3, SYSSFAO
      51 0C BC 3C 00015 MOVZWL   @RET_LEN, R1
      51 0C C4 00019 MULL2     #12, R1
      50 04 AC D0 0001C MOVL     SCB, R0
0200 C0 51 D1 00020 CMPL      R1, 512(R0)
      0C 15 00025 BLEQ       1$
      08 AC DD 00027 PUSHL     STR_DESC
      0C AC DD 0002A PUSHL     RET_LEN
      C8 AF 9F 0002D PUSHAB    SENT80_FORMAT
      62 03 FB 00030 CALLS     #3, SYSSFAO
      04 00033 1$: RET
```

; Routine Size: 52 bytes, Routine Base: CODE + 1CCC


```
3650 4566 1 %sbttl 'GET_EOF - Get the Phrase End of File'
3651 4567 1 ++
3652 4568 1 Functional Description:
3653 4569 1 This routine creates a phrase with 'EOF' or 'END OF FILE'.
3654 4570 1
3655 4571 1 Formal Parameters:
3656 4572 1     SCB          - Address of the SCB
3657 4573 1     STR_DESC    - Desc of String to Return
3658 4574 1     RET_LEN     - Return length of Desc.
3659 4575 1
3660 4576 1 Implicit Inputs:
3661 4577 1     none
3662 4578 1
3663 4579 1 Implicit Outputs:
3664 4580 1     none
3665 4581 1
3666 4582 1 Returned Value:
3667 4583 1     none
3668 4584 1
3669 4585 1 Side Effects:
3670 4586 1     none
3671 4587 1 --
3672 4588 1 ROUTINE GET_EOF (
3673 4589 1     SCB          : REF $BBLOCK,          ! SCB
3674 4590 1     STR_DESC     : REF VECTOR[2],        ! Output buffer desc
3675 4591 1     RET_LEN      : REF VECTOR [,WORD]    ! Return length (word)
3676 4592 1 ) : NOVALUE =
3677 4593 2 BEGIN
3678 4594 2 BIND
3679 4595 2     SENT132 FORMAT = $DESCRIPTOR (
3680 4596 2         'END OF FILE'),
3681 4597 2
3682 4598 2     SENT80 FORMAT = $DESCRIPTOR (
3683 4599 2         'EOF');
3684 4600 2
3685 4601 2 $FAO ( SENT132 FORMAT,
3686 4602 2     RET_LEN[0],          ! return length
3687 4603 2     STR_DESC[0],        ! address of string
3688 4604 2 );
3689 4605 2
3690 4606 2 ! Is it short enough to allow the words 'End of Job' to be printed ?
3691 4607 2
3692 4608 2 IF ((12 * .RET_LEN[0]) GTR .SCB[PSM$PAGE_WIDTH])
3693 4609 2 THEN
3694 4610 2     $FAO ( SENT80 FORMAT,
3695 4611 2         RET_LEN[0],          ! str[size] > fetched namelen
3696 4612 2         STR_DESC[0],
3697 4613 2     );
3698 4614 2
3699 4615 2 RETURN SS$NORMAL;
3700 4616 1 END;
```

```
45 4C 49 46 20 46 4F 20 44 4E 45 01D00 P.ACU: .ASCII \END OF FILE\
                                01D0B .BLKB 1
                                0000000B 01D0C P.ACT: .LONG 11
```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_EOF - Get the Phrase End of File

J 6
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 BLISS-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 122
(32)

00000000' 01D10 .ADDRESS P.ACU
46 4F 45 01D14 P.ACW: .ASCII \EOF\
01D17 .BLKB 1
00000003' 01D18 P.ACV: .LONG 3
00000000' 01D1C .ADDRESS P.ACW

SENT132_FORMAT= P.ACT
SENT80_FORMAT= P.ACV

52	00000000G	00	9E	00002	GET_EOF: .WORD	Save R2	4588
	08	AC	DD	00009	MOVAB	SYSSFAO, R2	
	0C	AC	DD	0000C	PUSHL	STR_DESC	4604
	DA	AF	9F	0000F	PUSHL	RET_LEN	
62		03	FB	00012	PUSHAB	SENT132_FORMAT	
51	0C	BC	3C	00015	CALLS	#3, SYSSFAO	
51		0C	C4	00019	MOVZWL	@RET_LEN, R1	4608
50	04	AC	D0	0001C	MULL2	#12, R1	
0200	C0	51	D1	00020	MOVL	SCB, R0	
		0C	15	00025	CMPL	R1, 512(R0)	
	08	AC	DD	00027	BLEQ	1\$	
	0C	AC	DD	0002A	PUSHL	STR_DESC	4613
	C8	AF	9F	0002D	PUSHL	RET_LEN	
62		03	FB	00030	PUSHAB	SENT80_FORMAT	
		04	00033	1\$:	CALLS	#3, SYSSFAO	
					RET		4616

; Routine Size: 52 bytes, Routine Base: CODE + 1D20

```
3702 4617 1 %sbttl 'GET_ACCOUNTING_INFO - Get the Accounting Information'
3703 4618 1 ++
3704 4619 1 Functional Description:
3705 4620 1 This routine returns a string containing the accounting information.
3706 4621 1
3707 4622 1 Formal Parameters:
3708 4623 1 SCB - Address of the SCB
3709 4624 1 STR_DESC - Desc of String to Return
3710 4625 1 RET_LEN - Return length of Desc.
3711 4626 1
3712 4627 1 Implicit Inputs:
3713 4628 1 none
3714 4629 1
3715 4630 1 Implicit Outputs:
3716 4631 1 none
3717 4632 1
3718 4633 1 Returned Value:
3719 4634 1 none
3720 4635 1
3721 4636 1 Side Effects:
3722 4637 1 none
3723 4638 1 --
3724 4639 1 ROUTINE GET_ACCOUNTING_INFO (
3725 4640 1 SCB : REF $BBLOCK, ! SCB
3726 4641 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
3727 4642 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
3728 4643 1 ) : NOVALUE =
3729 4644 2 BEGIN
3730 4645 2 BIND
3731 4646 2 SENT132 FORMAT = $DESCRIPTOR (
3732 4647 2 !#(AC)!,
3733 4648 2 !#(AS)!,
3734 4649 2 );
3735 4650 2 LOCAL
3736 4651 2 IF_PRES;
3737 4652 2
3738 4653 2 IF_PRES = .SCB_SIZE_ (ACCOUNTING_DATA);
3739 4654 2 IF .IF_PRES GEQ 1
3740 4655 2 THEN
3741 4656 2 IF_PRES = 1;
3742 4657 2
3743 4658 2 $FAO ( SENT132 FORMAT,
3744 4659 2 RET_LEN[0], ! return length
3745 4660 2 STR_DESC[0], ! address of string
3746 4661 2 .IF_PRES,
3747 4662 2 UPLIT BYTE (%ASCII 'ACCOUNTING INFO:'),
3748 4663 2 .IF_PRES,
3749 4664 2 SCB[PSM$Q_ACCOUNTING_DATA] ! accounting data
3750 4665 2 );
3751 4666 1 END;
```

```
29 43 41 28 23 21 01D54 P.ACY: .ASCII \!(AC)\
29 53 41 28 23 21 01D5A .ASCII \!(AS)\
0000000C 01D60 P.ACX: .LONG 12
00000000 01D64 .ADDRESS P.ACY
```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_ACCOUNTING_INFO - Get the Accounting Inform

L 6
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 124
(33)

46 4E 49 20 47 4E 49 54 4E 55 4F 43 43 41 10 01D68 P.ACZ: .ASCII <16>\ACCOUNTING INFO:\
3A 4F 01D77

SENT132_FORMAT= P.ACX

0000 00000 GET_ACCOUNTING_INFO:

50	04	AC	D0	00002	.WORD	Save nothing	4639
51	14	A0	3C	00006	MOVL	SCB, R0	4653
		03	15	0000A	MOVZWL	20(R0), IF_PRES	
51		01	D0	0000C	BLEQ	1\$	4654
	14	A0	9F	0000F	MOVL	#1, IF_PRES	4656
		51	DD	00012	PUSHAB	20(R0)-	4665
	D8	AF	9F	00014	PUSHL	IF_PRES	
		51	DD	00017	PUSHAB	P.ACZ	
	08	AC	DD	00019	PUSHL	IF_PRES	
	0C	AC	DD	0001C	PUSHL	STR_DESC	
	C5	AF	9F	0001F	PUSHL	RET_LEN	
00000000G 00		07	FB	00022	PUSHAB	SENT132_FORMAT	
		04	00029	CALLS	#7, SYS\$FAO		
				RET			4666

; Routine Size: 42 bytes, Routine Base: CODE + 1D79


```
3753 4667 1 %sbttl 'GET_QUALIFIERS - Get Switches/Qualifiers associated with PRINTING'
3754 4668 1 ++
3755 4669 1 Functional Description:
3756 4670 1 This routine returns a string containing the all relevant print
3757 4671 1 qualifier information.
3758 4672 1
3759 4673 1 Formal Parameters:
3760 4674 1 SCB - Address of the SCB
3761 4675 1 STR_DESC - Desc of String to Return
3762 4676 1 RET_LEN - Return length of Desc.
3763 4677 1
3764 4678 1 Implicit Inputs:
3765 4679 1 none
3766 4680 1
3767 4681 1 Implicit Outputs:
3768 4682 1 none
3769 4683 1
3770 4684 1 Returned Value:
3771 4685 1 none
3772 4686 1
3773 4687 1 Side Effects:
3774 4688 1 none
3775 4689 1 --
3776 4690 1 ROUTINE GET_QUALIFIERS (
3777 4691 1 SCB : REF $BLOCK, : SCB
3778 4692 1 STR_DESC : REF VECTOR[2], : Output buffer desc
3779 4693 1 RET_LEN : REF VECTOR [,WORD] : Return length (word)
3780 4694 1 ) : NOVALUE =
3781 4695 2 BEGIN
3782 4696 2 BIND
3783 4697 2 all the formats start here
3784 4698 2
3785 P 4699 2 DATE_FORMAT = $DESCRIPTOR (
3786 4700 2 '!'17XD'),
3787 4701 2
3788 P 4702 2 AFTER_TIME_FORMAT = $DESCRIPTOR (
3789 4703 2 '"/AFTER=!17XD'), ! - after_time print
3790 4704 2
3791 P 4705 2 BURST_FORMAT = $DESCRIPTOR (
3792 4706 2 '"/BURST'), ! - burst
3793 4707 2
3794 P 4708 2 CHARACTERISTICS_FORMAT = $DESCRIPTOR (
3795 4709 2 ' /CHARACTERISTICS=!AS'), ! - characteristics
3796 4710 2
3797 P 4711 2 FILE_COPIES_FORMAT = $DESCRIPTOR (
3798 4712 2 ' /COPIES=!UL(!UL of !UL)'), ! - copies/iteration
3799 4713 2
3800 P 4714 2 FEED_FORMAT = $DESCRIPTOR (
3801 4715 2 ' /FEED'), ! - feed
3802 4716 2
3803 P 4717 2 FLAG_FORMAT = $DESCRIPTOR (
3804 4718 2 ' /FLAG'), ! - flag
3805 4719 2
3806 P 4720 2 FORM_FORMAT = $DESCRIPTOR (
3807 4721 2 ' /FORM=!AS'), ! - form
3808 4722 2
3809 P 4723 2 HEADER_FORMAT = $DESCRIPTOR (
```

```
3810 4724 2      ' /HEADER'      ),      ! - header
3811 4725 2
3812 P 4726 2      JOB_COUNT_FORMAT = $DESCRIPTOR (
3813 4727 2      ' /JOB_COUNT=!UL(!UL of !UL)',      ! - job count /iteration
3814 4728 2
3815 P 4729 2      LENGTH_FORMAT = $DESCRIPTOR(
3816 4730 2      ' 7LENGTH=!UL'),
3817 4731 2
3818 P 4732 2      LIBRARY_FORMAT = $DESCRIPTOR (
3819 4733 2      ' /LIBRARY=!AS' ),      ! - library
3820 4734 2
3821 P 4735 2      MARGIN_FORMAT = $DESCRIPTOR(
3822 4736 2      ' 7MARGIN=('),
3823 4737 2
3824 P 4738 2      TOP_FORMAT = $DESCRIPTOR(
3825 4739 2      'TOP=!UL'),
3826 4740 2
3827 P 4741 2      BOTTOM_FORMAT = $DESCRIPTOR(
3828 4742 2      'BOTTOM=!UL'),
3829 4743 2
3830 P 4744 2      LEFT_FORMAT = $DESCRIPTOR(
3831 4745 2      'LEFT=!UL'),
3832 4746 2
3833 P 4747 2      RIGHT_FORMAT = $DESCRIPTOR(
3834 4748 2      'RIGHT=!UL'),
3835 4749 2
3836 P 4750 2      NOFEED_FORMAT = $DESCRIPTOR (
3837 4751 2      ' 7NOFEED'      ),      ! - nofeed
3838 4752 2
3839 P 4753 2      SETUP_PAGE_FORMAT = $DESCRIPTOR(
3840 4754 2      ' /PAGE_SETUP=(!AS)'      ),      ! - setup page
3841 4755 2
3842 P 4756 2      PAGES_FORMAT = $DESCRIPTOR (
3843 4757 2      ' /PAGES=(!UL,!UL)'      ),      ! - page count
3844 4758 2
3845 P 4759 2      PARAMETER_FORMAT = $DESCRIPTOR (
3846 4760 2      ' /PARAMETERS=(
3847 4761 2      '...!AS', '!AS', '!AS', '!AS', '!AS', ' ,
3848 4762 2      '...!AS', '!AS', '!AS'),
3849 4763 2      ! - parameter lists
3850 P 4764 2      PASSALL_FORMAT = $DESCRIPTOR (
3851 4765 2      ' /PASSALL'      ),      ! - passall
3852 4766 2
3853 P 4767 2      PUNCTUATION_FORMAT = $DESCRIPTOR(
3854 4768 2      ' !AC'),      ! comma or close paren
3855 4769 2
3856 P 4770 2      SETUP_FILE_FORMAT = $DESCRIPTOR (
3857 4771 2      ' /SETUP_FILE=(!AS)'      ),      ! - setup file
3858 4772 2
3859 P 4773 2      SETUP_FORM_FORMAT = $DESCRIPTOR(
3860 4774 2      ' /SETUP_FORM=(!AS)'      ),      ! - setup form
3861 4775 2
3862 P 4776 2      SHEET_FORMAT = $DESCRIPTOR(
3863 4777 2      ' /SHEET_FEED'      ),
3864 4778 2
3865 P 4779 2      SPACE_FORMAT = $DESCRIPTOR (
3866 4780 2      ' /SPACE'      ),      ! - space
```

```
3867 4781 2
3868 P 4782 2 TRAILER_FORMAT = $DESCRIPTOR (
3869 4783 2 ' /TRAILER' ), ! - trailer
3870 4784 2
3871 P 4785 2 TRUNCATE_FORMAT = $DESCRIPTOR(
3872 4786 2 ' /TRUNCATE' ),
3873 4787 2
3874 P 4788 2 WIDTH_FORMAT = $DESCRIPTOR(
3875 4789 2 ' /WIDTH=!UL' ),
3876 4790 2
3877 P 4791 2 WRAP_FORMAT = $DESCRIPTOR(
3878 4792 2 ' /WRAP' );
3879 4793 2
3880 4794 2 LITERAL
3881 4795 2 K_MAX_BUFFER_SIZE = 512;
3882 4796 2
3883 4797 2 LOCAL
3884 4798 2 PUNC_FLAG : INITIAL (0),
3885 4799 2 TEMP_LEN
3886 4800 2 AFT_DATE_PTR: VECTOR[2],
3887 4801 2 TEMP_PTR : VECTOR[2],
3888 4802 2 AFT_BUFF : VECTOR[17,byte],
3889 4803 2 TEMP_BUFF : VECTOR[17,byte],
3890 4804 2 IF PRES
3891 4805 2 CURRENT_LEN : INITIAL (0),
3892 4806 2 STRING_PTR : VECTOR [2]; ! Pointer to current string
3893 4807 2
3894 4808 2 ! Allocate the buffer for "GET_xxx" Routines
3895 4809 2
3896 4810 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
3897 4811 2 STRING_PTR[ADDR] = .STR_DESC[ADDR]; ! init address
3898 4812 2
3899 4813 2 RET_LEN[0] = 0;
3900 4814 2
3901 4815 2 $FAO ( BEGIN_FORMAT,
3902 4816 2 CURRENT_LEN, ! return length
3903 4817 2 STRING_PTR[0], ! address of string
3904 4818 2 ! );
3905 4819 2
3906 4820 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3907 4821 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3908 4822 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3909 4823 2
3910 4824 2
3911 4825 2 ! Increment pointer only if not equal to time queued
3912 4826 2
3913 4827 2 AFT_DATE_PTR[SIZE] = %ALLOCATION(AFT_BUFF);
3914 4828 2 AFT_DATE_PTR[ADDR] = AFT_BUFF;
3915 4829 2
3916 P 4830 2 $FAO ( DATE_FORMAT,
3917 4831 2 TEMP_LEN,
3918 4832 2 AFT_DATE_PTR[0],
3919 4833 2 SCB[PSM$0_AFTER_TIME]);
3920 4834 2
3921 4835 2 TEMP_PTR[SIZE] = %ALLOCATION(TEMP_BUFF);
3922 4836 2 TEMP_PTR[ADDR] = TEMP_BUFF;
3923 4837 2
```

```
.. 3924 P 4838 2 $FAO ( DATE_FORMAT,  
.. 3925 P 4839 TEMP_LEN,  
.. 3926 P 4840 TEMP_PTR[0],  
.. 3927 4841 SCB[PSM$Q_TIME_QUEUED]);  
.. 3928 4842  
.. 3929 4843 IF CH$NEQ( .TEMP_LEN, .TEMP_PTR[ADDR], .TEMP_LEN, .AFT_DATE_PTR[ADDR])  
.. 3930 4844 THEN  
.. 3931 4845 BEGIN  
.. 3932 4846  
.. 3933 P 4847 $FAO (  
.. 3934 P 4848 AFTER TIME FORMAT,  
.. 3935 P 4849 CURRENT_LEN, ! return length  
.. 3936 P 4850 STRING_PTR[0], ! address of string  
.. 3937 4851 SCB[PSM$Q_AFTER_TIME]); ! after_time  
.. 3938 4852  
.. 3939 4853 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;  
.. 3940 4854 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;  
.. 3941 4855 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];  
.. 3942 4856 END;  
.. 3943 4857  
.. 3944 4858 IF .SEPARATE_FLAG_ (FILE_BURST)  
.. 3945 4859 THEN  
.. 3946 4860 BEGIN  
.. 3947 P 4861 $FAO ( BURST FORMAT,  
.. 3948 P 4862 CURRENT_LEN, ! return length  
.. 3949 P 4863 STRING_PTR[0] ! address of string  
.. 3950 4864 );  
.. 3951 4865  
.. 3952 4866 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;  
.. 3953 4867 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;  
.. 3954 4868 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];  
.. 3955 4869 END;  
.. 3956 4870  
.. 3957 4871 IF (.SCB_SIZE_ (CHARACTERISTICS) EQL 0)  
.. 3958 4872 THEN  
.. 3959 4873 BEGIN  
.. 3960 P 4874 $FAO ( CHARACTERISTICS_FORMAT,  
.. 3961 P 4875 CURRENT_LEN, ! return length  
.. 3962 P 4876 STRING_PTR[0], ! address of string  
.. 3963 P 4877 SCB[PSM$Q_CHARACTERISTICS] ! /CHARACTERISTICS  
.. 3964 4878 );  
.. 3965 4879  
.. 3966 4880 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;  
.. 3967 4881 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;  
.. 3968 4882 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];  
.. 3969 4883 END;  
.. 3970 4884  
.. 3971 4885 ! Always print something about form feed... /FEED or /NOFEED  
.. 3972 4886  
.. 3973 4887 IF .SBBLOCK[SCB[PSM$L_PRINT_CONTROL], SMBMSG$V_PAGINATE]  
.. 3974 4888 THEN  
.. 3975 4889 BEGIN  
.. 3976 P 4890 $FAO ( FEED FORMAT,  
.. 3977 P 4891 CURRENT_LEN, ! return length  
.. 3978 P 4892 STRING_PTR[0] ! address of string  
.. 3979 4893 );  
.. 3980 4894
```



```
3981 4895 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3982 4896 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3983 4897 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3984 4898 3 END;
3985 4899 3
3986 4900 3 IF .SCB[PSM$L_FILE_COPIES] GTR 1
3987 4901 3 THEN
3988 4902 3 BEGIN
3989 4903 3 $FAO ( FILE_COPIES_FORMAT,
3990 4904 3 CURRENT_LEN, ! return length
3991 4905 3 STRING_PTR[0], ! address of string
3992 4906 3 .SCB[PSM$L_FILE_COPIES], ! /FILE_COPIES
3993 4907 3 .SCB[PSM$L_FILE_COUNT], ! iteration
3994 4908 3 .SCB[PSM$L_FILE_COPIES]
3995 4909 3 );
3996 4910 3
3997 4911 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3998 4912 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3999 4913 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4000 4914 3 END;
4001 4915 3
4002 4916 3 IF .SEPARATE_FLAG_ (FILE_FLAG)
4003 4917 3 THEN
4004 4918 3 BEGIN
4005 4919 3 $FAO ( FLAG_FORMAT,
4006 4920 3 CURRENT_LEN, ! return length
4007 4921 3 STRING_PTR[0] ! address of string
4008 4922 3 );
4009 4923 3
4010 4924 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4011 4925 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4012 4926 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4013 4927 3 END;
4014 4928 3
4015 4929 3 $FAO ( FORM_FORMAT,
4016 4930 3 CURRENT_LEN, ! return length
4017 4931 3 STRING_PTR[0], ! address of string
4018 4932 3 SCB[PSM$Q_FORM_NAME]);
4019 4933 3
4020 4934 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4021 4935 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4022 4936 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4023 4937 3
4024 4938 3 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4025 4939 3 THEN
4026 4940 3 BEGIN
4027 4941 3 RET_LEN[0] = 512;
4028 4942 3 RETURN;
4029 4943 3 END;
4030 4944 3
4031 4945 3 IF .SBBLOCK[SCB[PSM$L_PRINT_CONTROL], SMBMSG$V_PAGE_HEADER]
4032 4946 3 THEN
4033 4947 3 BEGIN
4034 4948 3 $FAO ( HEADER_FORMAT,
4035 4949 3 CURRENT_LEN, ! return length
4036 4950 3 STRING_PTR[0] ! address of string
4037 4951 3 );
```

```
4038 4952 3
4039 4953 RET LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4040 4954 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4041 4955 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4042 4956 END;
4043 4957
4044 4958 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4045 4959 THEN
4046 4960 BEGIN
4047 4961 RET_LEN[0] = 512;
4048 4962 RETURN;
4049 4963 END;
4050 4964
4051 4965 IF .SCB[PSMSL_JOB_COPIES] GTR 1
4052 4966 THEN
4053 4967 BEGIN
4054 4968 $FAO ( JOB COUNT FORMAT,
P 4969 CURRENT_LEN,
P 4970 STRING_PTR[0],
P 4971 .SCB[PSMSL_JOB_COPIES],
P 4972 .SCB[PSMSL_JOB_COUNT],
P 4973 .SCB[PSMSL_JOB_COPIES]
4060 4974 );
4061 4975
4062 4976 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4063 4977 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4064 4978 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4065 4979 END;
4066 4980
4067 4981 ! Here is my internal call to PSMSREAD_ITEM_DX to insure that the user
4068 4982 ! can copy information successfully using this routine
4069 4983
4070 4984 BEGIN
4071 4985 LOCAL LEN: VECTOR[2];
4072 4986
4073 4987 INIT_DYN_DESC (LEN);
4074 4988 PSMSREAD_ITEM_DX (.SCB, %ref(SMBMSG$K_FORM_LENGTH),
4075 4989 LEN[0]);
4076 4990
4077 4991 $FAO ( LENGTH FORMAT,
P 4992 CURRENT_LEN,
P 4993 STRING_PTR[0],
P 4994 ..LEN[ADDR]
4081 4995 );
4082 4996
4083 4997 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4084 4998 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4085 4999 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4086 5000 END;
4087 5001
4088 5002 !*** ALWAYS PRINT THE LIBRARY !**!
4089 5003
4090 5004 $FAO ( LIBRARY FORMAT,
P 5005 CURRENT_LEN,
P 5006 STRING_PTR[0],
P 5007 SCB[PSMSQ_LIBRARY_SPECIFICATION]
4094 5008 );
```

```
! return length
! address of string
! /JOB_COUNT
! iteration
```

```
! return length
! address of string
! length pointed to by len[addr]
```

```
! return length
! address of string
! /LIBRARY
```

```
4095 5009 2
4096 5010 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4097 5011 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4098 5012 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4099 5013 2
4100 5014 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4101 5015 2 THEN
4102 5016 2 BEGIN
4103 5017 2     RET_LEN[0] = 512;
4104 5018 2     RETURN;
4105 5019 2 END;
4106 5020 2
4107 5021 2 IF (.SCB[PSMSL_TOP_MARGIN] NEQ 0) OR
4108 5022 2     (.SCB[PSMSL_BOTTOM_MARGIN] NEQ 0) OR
4109 5023 2     (.SCB[PSMSL_LEFT_MARGIN] NEQ 0) OR
4110 5024 2     (.SCB[PSMSL_RIGHT_MARGIN] NEQ 0)
4111 5025 2 THEN
4112 5026 2 BEGIN
4113 5027 2
4114 5028 2     $FAO ( MARGIN_FORMAT,
P 5029 2         CURRENT_LEN,                ! return length
P 5030 2         STRING_PTR[0],              ! address of string
4117 5031 2     );
4118 5032 2
4119 5033 2     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4120 5034 2     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4121 5035 2     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4122 5036 2
4123 5037 2 IF (.SCB[PSMSL_TOP_MARGIN] NEQ 0)
4124 5038 2 THEN
4125 5039 2 BEGIN
4126 5040 2     $FAO ( TOP_FORMAT,
P 5041 2         CURRENT_LEN,                ! return length
P 5042 2         STRING_PTR[0],              ! address of string
P 5043 2         .SCB[PSMSL_TOP_MARGIN],    ! top
4130 5044 2     );
4131 5045 2
4132 5046 2     PUNC_FLAG = 1;
4133 5047 2     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4134 5048 2     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4135 5049 2     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4136 5050 2 END;
4137 5051 2
4138 5052 2 IF (.SCB[PSMSL_BOTTOM_MARGIN] NEQ 0) AND
4139 5053 2     .PUNC_FLAG
4140 5054 2 THEN
4141 5055 2 BEGIN
4142 5056 2     $FAO(
P 5057 2         PUNCTUATION_FORMAT,
P 5058 2         CURRENT_LEN,                ! return length
P 5059 2         STRING_PTR[0],              ! address of string
4146 5060 2         UPLIT BYTE (%ASCII ',')
4147 5061 2     );
4148 5062 2
4149 5063 2     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4150 5064 2     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4151 5065 2     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
```

```
4152 5066 3      END;
4153 5067 3
4154 5068 4      IF (.SCB[PSMSL_BOTTOM_MARGIN] NEQ 0)
4155 5069 3      THEN
4156 5070 4      BEGIN
4157 P 5071 4      $FAO (
4158 P P 5072 4      BOTTOM FORMAT,
4159 P P 5073 4      CURRENT_LEN,           ! return length
4160 P P 5074 4      STRING_PTR[0],         ! address of string
4161 P 5075 4      ,SCB[PSMSL_BOTTOM_MARGIN] ! bottom
4162 5076 4      );
4163 5077 4
4164 5078 4      PUNC_FLAG = 1;
4165 5079 4      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4166 5080 4      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4167 5081 4      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4168 5082 3      END;
4169 5083 3
4170 5084 3      IF (.SCB[PSMSL_LEFT_MARGIN] NEQ 0) AND
4171 5085 3      .PUNC_FLAG
4172 5086 3      THEN
4173 5087 4      BEGIN
4174 P 5088 4      $FAO(
4175 P P 5089 4      PUNCTUATION_FORMAT,
4176 P P 5090 4      CURRENT_LEN,           ! return length
4177 P P 5091 4      STRING_PTR[0],         ! address of string
4178 P 5092 4      UPLIT BYTE (XASCIC ',')
4179 5093 4      );
4180 5094 4      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4181 5095 4      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4182 5096 4      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4183 5097 3      END;
4184 5098 3
4185 5099 4      IF (.SCB[PSMSL_LEFT_MARGIN] NEQ 0)
4186 5100 3      THEN
4187 5101 4      BEGIN
4188 P 5102 4      $FAO ( LEFT FORMAT,
4189 P P 5103 4      CURRENT_LEN,           ! return length
4190 P P 5104 4      STRING_PTR[0],         ! address of string
4191 P 5105 4      ,SCB[PSMSL_LEFT_MARGIN] ! left
4192 5106 4      );
4193 5107 4
4194 5108 4      PUNC_FLAG = 1;
4195 5109 4      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4196 5110 4      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4197 5111 4      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4198 5112 3      END;
4199 5113 3
4200 5114 3      IF (.SCB[PSMSL_RIGHT_MARGIN] NEQ 0) AND
4201 5115 3      .PUNC_FLAG
4202 5116 3      THEN
4203 5117 4      BEGIN
4204 P 5118 4      $FAO(
4205 P P 5119 4      PUNCTUATION_FORMAT,
4206 P P 5120 4      CURRENT_LEN,           ! return length
4207 P 5121 4      STRING_PTR[0],         ! address of string
4208 P 5122 4      UPLIT BYTE (XASCIC ',')
```



```

: 4209      5123 4      );
: 4210      5124 4
: 4211      5125 4      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4212      5126 4      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4213      5127 4      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4214      5128 3      END;
: 4215      5129 3
: 4216      5130 4      IF (.SCB[PSMSL_RIGHT_MARGIN] NEQ 0)
: 4217      5131 3      THEN
: 4218      5132 4      BEGIN
: 4219      P 5133 4      $FAO ( RIGHT FORMAT,
: 4220      P 5134 4      CURRENT_LEN,          ! return length
: 4221      P 5135 4      STRING_PTR[0],        ! address of string
: 4222      P 5136 4      .SCB[PSMSL_RIGHT_MARGIN], ! right
: 4223      5137 4      );
: 4224      5138 4
: 4225      5139 4      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4226      5140 4      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4227      5141 4      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4228      5142 3      END;
: 4229      5143 3
: 4230      P 5144 3      $FAO(
: 4231      P 5145 3      PUNCTUATION_FORMAT,
: 4232      P 5146 3      CURRENT_LEN,          ! return length
: 4233      P 5147 3      STRING_PTR[0],        ! address of string
: 4234      P 5148 3      UPLIT BYTE ('ASCII '))
: 4235      5149 3      );
: 4236      5150 3
: 4237      5151 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4238      5152 3      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4239      5153 3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4240      5154 2      END;
: 4241      5155 2
: 4242      5156 2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4243      5157 2      THEN
: 4244      5158 2      BEGIN
: 4245      5159 2      RET_LEN[0] = 512;
: 4246      5160 2      RETURN;
: 4247      5161 2      END;
: 4248      5162 2
: 4249      5163 3      IF NOT (.SBBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_PAGINATE])
: 4250      5164 2      THEN
: 4251      5165 2      BEGIN
: 4252      P 5166 2      $FAO ( NOFEED FORMAT,
: 4253      P 5167 2      CURRENT_LEN,          ! return length
: 4254      P 5168 2      STRING_PTR[0]        ! address of string
: 4255      5169 2      );
: 4256      5170 2
: 4257      5171 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4258      5172 3      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4259      5173 3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4260      5174 2      END;
: 4261      5175 2
: 4262      5176 2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4263      5177 2      THEN
: 4264      5178 2      BEGIN
: 4265      5179 3      RET_LEN[0] = 512;
```

```
.. 4266      5180      3      RETURN;
.. 4267      5181      2      END;
.. 4268      5182
.. 4269      5183      IF (.SCB[PSMSL_FIRST_PAGE] NEQ 0) OR      ! default last page is zero
.. 4270      5184      (.SCB[PSMSL_LAST_PAGE] NEQ 0)
.. 4271      5185      THEN
.. 4272      5186      BEGIN
.. 4273      5187      $FAO (      PAGES FORMAT,
.. 4274      5188      CURRENT_LEN,      ! return length
.. 4275      5189      STRING_PTR[0],      ! address of string
.. 4276      5190      .SCB[PSMSL_FIRST_PAGE],
.. 4277      5191      .SCB[PSMSL_LAST_PAGE]
.. 4278      5192      );
.. 4279      5193
.. 4280      5194      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
.. 4281      5195      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
.. 4282      5196      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
.. 4283      5197      END;
.. 4284      5198
.. 4285      5199      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
.. 4286      5200      THEN
.. 4287      5201      BEGIN
.. 4288      5202      RET_LEN[0] = 512;
.. 4289      5203      RETURN;
.. 4290      5204      END;
.. 4291      5205
.. 4292      5206      IF (.SCB_SIZE_ (PARAMETER_1) NEQ 0 OR
.. 4293      5207      .SCB_SIZE_ (PARAMETER_2) NEQ 0 OR
.. 4294      5208      .SCB_SIZE_ (PARAMETER_3) NEQ 0 OR
.. 4295      5209      .SCB_SIZE_ (PARAMETER_4) NEQ 0 OR
.. 4296      5210      .SCB_SIZE_ (PARAMETER_5) NEQ 0 OR
.. 4297      5211      .SCB_SIZE_ (PARAMETER_6) NEQ 0 OR
.. 4298      5212      .SCB_SIZE_ (PARAMETER_7) NEQ 0 OR
.. 4299      5213      .SCB_SIZE_ (PARAMETER_8) NEQ 0 )
.. 4300      5214      THEN
.. 4301      5215      BEGIN
.. 4302      5216      $FAO (
.. 4303      5217      PARAMETER FORMAT,
.. 4304      5218      CURRENT_LEN,      ! return length
.. 4305      5219      STRING_PTR[0],      ! address of string
.. 4306      5220      SCB[PSMSQ_PARAMETER_1],      P1
.. 4307      5221      SCB[PSMSQ_PARAMETER_2],      P2
.. 4308      5222      SCB[PSMSQ_PARAMETER_3],      P3
.. 4309      5223      SCB[PSMSQ_PARAMETER_4],      P4
.. 4310      5224      SCB[PSMSQ_PARAMETER_5],      P5
.. 4311      5225      SCB[PSMSQ_PARAMETER_6],      P6
.. 4312      5226      SCB[PSMSQ_PARAMETER_7],      P7
.. 4313      5227      SCB[PSMSQ_PARAMETER_8],      P8
.. 4314      5228      );
.. 4315      5229
.. 4316      5230      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
.. 4317      5231      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
.. 4318      5232      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
.. 4319      5233      END;
.. 4320      5234
.. 4321      5235      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
.. 4322      5236      THEN
```

```
4323 5237 3 BEGIN
4324 5238 RET LEN[0] = 512;
4325 5239 RETURN;
4326 5240 END;
4327 5241
4328 5242 IF .SBBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_PASSALL]
4329 5243 THEN
4330 5244 BEGIN
4331 5245 $FAO ( PASSALL_FORMAT,
4332 5246 CURRENT_LEN, ! return length
4333 5247 STRING_PTR[0], ! address of string
4334 5248 );
4335 5249
4336 5250 RET LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4337 5251 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4338 5252 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4339 5253 END;
4340 5254
4341 5255 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4342 5256 THEN
4343 5257 BEGIN
4344 5258 RET LEN[0] = 512;
4345 5259 RETURN;
4346 5260 END;
4347 5261
4348 5262 IF .SCB_SIZE_ (FILE_SETUP_MODULES) GTR 0
4349 5263 THEN
4350 5264 BEGIN
4351 5265 $FAO ( SETUP_FILE_FORMAT,
4352 5266 CURRENT_LEN, ! return length
4353 5267 STRING_PTR[0], ! address of string
4354 5268 SCB[PSMSQ_FILE_SETUP_MODULES]
4355 5269 );
4356 5270
4357 5271 RET LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4358 5272 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4359 5273 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4360 5274 END;
4361 5275
4362 5276 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4363 5277 THEN
4364 5278 BEGIN
4365 5279 RET LEN[0] = 512;
4366 5280 RETURN;
4367 5281 END;
4368 5282
4369 5283 IF .SCB_SIZE_ (FORM_SETUP_MODULES) GTR 0
4370 5284 THEN
4371 5285 BEGIN
4372 5286 $FAO ( SETUP_FORM_FORMAT,
4373 5287 CURRENT_LEN, ! return length
4374 5288 STRING_PTR[0], ! address of string
4375 5289 SCB[PSMSQ_FORM_SETUP_MODULES] ! form setup
4376 5290 );
4377 5291
4378 5292 RET LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4379 5293 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
```

```

: 4380      5294      3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4381      5295      END;
: 4382      5296
: 4383      5297      IF .SCB_SIZE_ (PAGE_SETUP_MODULES) GTR 0
: 4384      5298      THEN
: 4385      5299      BEGIN
: 4386      5300      $FAO (  SETUP PAGE FORMAT,
: 4387      5301      CURRENT_LEN,
: 4388      5302      STRING_PTR[0],
: 4389      5303      SCB[PSMSQ_PAGE_SETUP_MODULES]
: 4390      5304      );
: 4391      5305
: 4392      5306      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4393      5307      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4394      5308      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4395      5309      END;
: 4396      5310
: 4397      5311      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4398      5312      THEN
: 4399      5313      BEGIN
: 4400      5314      RET_LEN[0] = 512;
: 4401      5315      RETURN;
: 4402      5316      END;
: 4403      5317
: 4404      5318      IF .$BLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_SHEET_FEED]
: 4405      5319      THEN
: 4406      5320      BEGIN
: 4407      5321      $FAO (  SHEET FORMAT,
: 4408      5322      CURRENT_LEN,
: 4409      5323      STRING_PTR[0]
: 4410      5324      );
: 4411      5325
: 4412      5326      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4413      5327      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4414      5328      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4415      5329      END;
: 4416      5330
: 4417      5331      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4418      5332      THEN
: 4419      5333      BEGIN
: 4420      5334      RET_LEN[0] = 512;
: 4421      5335      RETURN;
: 4422      5336      END;
: 4423      5337
: 4424      5338      IF .$BLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_DOUBLE_SPACE]
: 4425      5339      THEN
: 4426      5340      BEGIN
: 4427      5341      $FAO (  SPACE FORMAT,
: 4428      5342      CURRENT_LEN,
: 4429      5343      STRING_PTR[0]
: 4430      5344      );
: 4431      5345
: 4432      5346      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4433      5347      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4434      5348      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4435      5349      END;
: 4436      5350
```



```
4437 5351 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4438 5352 THEN
4439 5353 BEGIN
4440 5354     RET_LEN[0] = 512;
4441 5355     RETURN;
4442 5356 END;
4443 5357
4444 5358 IF .SEPARATE_FLAG_ (FILE_TRAILER)
4445 5359 THEN
4446 5360 BEGIN
4447 5361     $FAO ( TRAILER_FORMAT,
4448 5362           CURRENT_LEN,           ! return length
4449 5363           STRING_PTR[0]         ! address of string
4450 5364     );
4451 5365
4452 5366     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4453 5367     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4454 5368     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4455 5369 END;
4456 5370
4457 5371 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4458 5372 THEN
4459 5373 BEGIN
4460 5374     RET_LEN[0] = 512;
4461 5375     RETURN;
4462 5376 END;
4463 5377
4464 5378 IF .$BLOCK[SCB[PSM$L_PRINT_CONTROL], SMBMSG$V_TRUNCATE]
4465 5379 THEN
4466 5380 BEGIN
4467 5381     $FAO ( TRUNCATE_FORMAT,
4468 5382           CURRENT_LEN,           ! return length
4469 5383           STRING_PTR[0]         ! address of string
4470 5384     );
4471 5385
4472 5386     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4473 5387     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4474 5388     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4475 5389 END;
4476 5390
4477 5391 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4478 5392 THEN
4479 5393 BEGIN
4480 5394     RET_LEN[0] = 512;
4481 5395     RETURN;
4482 5396 END;
4483 5397
4484 5398 $FAO ( WIDTH_FORMAT,
4485 5399       CURRENT_LEN,           ! return length
4486 5400       STRING_PTR[0],         ! address of string
4487 5401       .SCB[PSM$L_FORM_WIDTH] ! form width
4488 5402     );
4489 5403
4490 5404     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4491 5405     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4492 5406     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4493 5407
```

```

4494      5408      2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4495      5409      THEN
4496      5410      BEGIN
4497      5411      RET_LEN[0] = 512;
4498      5412      RETURN;
4499      5413      END;
4500      5414
4501      5415      IF .SBBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_WRAP]
4502      5416      THEN
4503      5417      BEGIN
4504      5418      $FAO ( WRAP FORMAT,
4505      5419      CURRENT_LEN,
4506      5420      STRING_PTR[0]
4507      5421      );
4508      5422
4509      5423      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4510      5424      END;
4511      5425
4512      5426      ! Don't print anything if no flags were set
4513      5427      !
4514      5428      IF .RET_LEN[0] LEQ 18
4515      5429      THEN
4516      5430      RET_LEN[0] = 0;
4517      5431
4518      5432      ! Length returned must be less than max string size
4519      5433      !
4520      5434      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4521      5435      THEN
4522      5436      BEGIN
4523      5437      RET_LEN[0] = 512;
4524      5438      RETURN;
4525      5439      END;
4526      5440
4527      5441      1 END;

```

44	25	37	31	21	3D	52	45	54	46	41	2F	20	01DA3	P.ADB:	.ASCII	\!17XD\		
													00000005	01DA8	P.ADA:	.LONG	5	
													00000000	01DAC		.ADDRESS	P.ADB	
44	25	37	31	21	3D	52	45	54	46	41	2F	20	01DB0	P.ADD:	.ASCII	\ /AFTER=!17XD\		
													01DBD		.BLKB	3		
													0000000D	01DC0	P.ADC:	.LONG	13	
													00000000	01DC4		.ADDRESS	P.ADD	
						54	53	52	55	42	2F	20	01DC8	P.ADF:	.ASCII	\ /BURST\		
													01DCF		.BLKB	1		
													00000007	01DD0	P.ADE:	.LONG	7	
													00000000	01DD4		.ADDRESS	P.ADF	
49	54	53	49	52	45	54	43	41	52	41	48	43	2F	20	01DD8	P.ADH:	.ASCII	\ /CHARACTERISTICS=!AS\
									53	41	21	3D	53	43	01DE7			
															01DED		.BLKB	3
													00000015	01DF0	P.ADG:	.LONG	21	
													00000000	01DF4		.ADDRESS	P.ADH	
55	21	28	4C	55	21	3D	53	45	49	50	4F	43	2F	20	01DF8	P.ADJ:	.ASCII	\ /COPIES=!UL(!UL of !UL)\
						29	4C	55	21	20	66	6F	20	4C	01E07			
															01E10	P.ADI:	.LONG	24
													00000018	01E14		.ADDRESS	P.ADJ	
													00000000					

```

      44 45 45 46 2F 20 01E18 P.ADL: .ASCII \ /FEED\
                        01E1E .BLKB 2
                        00000006 01E20 P.ADK: .LONG 6
                        00000000 01E24 .ADDRESS P.ADL
      47 41 4C 46 2F 20 01E28 P.ADN: .ASCII \ /FLAG\
                        01E2E .BLKB 2
                        00000006 01E30 P.ADM: .LONG 6
                        00000000 01E34 .ADDRESS P.ADN
      53 41 21 3D 4D 52 4F 46 2F 20 01E38 P.ADP: .ASCII \ /FORM=!AS\
                        01E42 .BLKB 2
                        0000000A 01E44 P.ADO: .LONG 10
                        00000000 01E48 .ADDRESS P.ADP
      52 45 44 41 45 48 2F 20 01E4C P.ADR: .ASCII \ /HEADER\
                        00000008 01E54 P.ADQ: .LONG 8
                        00000000 01E58 .ADDRESS P.ADR
4C 55 21 3D 54 4E 55 4F 43 5F 42 4F 4A 2F 20 01E5C P.ADT: .ASCII \ /JOB_COUNT=!UL(!UL of !UL)\
29 4C 55 21 20 66 6F 20 4C 55 21 28 01E6B .BLKB 1
                        01E77 .LONG 27
                        0000001B 01E78 P.ADS: .ADDRESS P.ADT
                        00000000 01E7C .ASCII \ /LENGTH=!UL\
      4C 55 21 3D 48 54 47 4E 45 4C 2F 20 01E80 P.ADV: .LONG 12
                        0000000C 01E8C P.ADU: .ADDRESS P.ADV
                        00000000 01E90 .ASCII \ /LIBRARY=!AS\
      53 41 21 3D 59 52 41 52 42 49 4C 2F 20 01E94 P.ADX: .BLKB 3
                        01EA1 .LONG 13
                        0000000D 01EA4 P.ADW: .ADDRESS P.ADX
                        00000000 01EA8 .ASCII \ /MARGIN=(\
      28 3D 4E 49 47 52 41 4D 2F 20 01EAC P.ADZ: .BLKB 2
                        01EB6 .LONG 10
                        0000C00A 01EB8 P.ADY: .ADDRESS P.ADZ
                        00000000 01EBC .ASCII \TOP=!UL\
      4C 55 21 3D 50 4F 54 01EC0 P.AEB: .BLKB 1
                        01EC7 .LONG 7
                        00000007 01EC8 P.AEA: .ADDRESS P.AEB
                        00000000 01ECC .ASCII \BOTTOM=!UL\
      4C 55 21 3D 4D 4F 54 54 4F 42 01ED0 P.AED: .BLKB 2
                        01EDA .LONG 10
                        0000000A 01EDC P.AEC: .ADDRESS P.AED
                        00000000 01EE0 .ASCII \LEFT=!UL\
      4C 55 21 3D 54 46 45 4C 01EE4 P.AEF: .LONG 8
                        00000008 01EEC P.AEE: .ADDRESS P.AEF
                        00000000 01EF0 .ASCII \RIGHT=!UL\
      4C 55 21 3D 54 48 47 49 52 01EF4 P.AEH: .BLKB 3
                        01EFD .LONG 9
                        00000009 01F00 P.AEG: .ADDRESS P.AEH
                        00000000 01F04 .ASCII \ /NOFEED\
      44 45 45 46 4F 4E 2F 20 01F08 P.AEJ: .LONG 8
                        00000008 01F10 P.AEI: .ADDRESS P.AEJ
                        00000000 01F14 .ASCII \ /PAGE_SETUP=(!AS)\
21 28 3D 50 55 54 45 53 5F 45 47 41 50 2F 20 01F18 P.AEL: .BLKB 2
29 53 41 01F27 .LONG 18
                        01F2A .ADDRESS P.AEL
                        00000012 01F2C P.AEK: .ASCII \ /PAGES=(!UL,!UL)\
                        00000000 01F30 .BLKB 3
55 21 2C 4C 55 21 28 3D 53 45 47 41 50 2F 20 01F34 P.AEN:
29 4C 01F43
01F45
```

SEP
V04

[illegible]

DATE FORMAT= P.ADA
AFTER TIME FORMAT= P.ADC
BURST FORMAT= P.ADE
CHARACTERISTICS_FORMAT= P.ADG
FILE_COPIES_FORMAT= P.ADI
FEED_FORMAT= P.ADK
FLAG_FORMAT= P.ADM
FORM_FORMAT= P.ADO
HEADER_FORMAT= P.ADQ
JOB COUNT FORMAT= P.ADS
LENGTH FORMAT= P.ADU
LIBRARY_FORMAT= P.ADW
MARGIN_FORMAT= P.ADY
TOP_FORMAT= P.AEA
BOTTOM_FORMAT= P.AEC
LEFT_FORMAT= P.AEE
RIGHT_FORMAT= P.AEG
NOFEED_FORMAT= P.AEI
SETUP_PAGE_FORMAT= P.AEK
PAGES_FORMAT= P.AEM
PARAMETER_FORMAT= P.AEO
PASSALL_FORMAT= P.AEQ
PUNCTUATION_FORMAT= P.AES
SETUP_FILE_FORMAT= P.AEU
SETUP_FORM_FORMAT= P.AEW
SHEET_FORMAT= P.AEY
SPACE_FORMAT= P.AFA
TRAILER_FORMAT= P.AFC
TRUNCATE_FORMAT= P.AFE
WIDTH_FORMAT= P.AFG
WRAP_FORMAT= P.AFI

07FC 00000 GET_QUALIFIERS:

5A	FF42	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10	4690
59	00000000G	00	9E	00007	MOVAB	PUNCTUATION_FORMAT, R10	
5E	AC	AE	9E	0000E	MOVAB	SYSSFAO, R9	
	08	56	D4	00012	MOVAB	-84(SP), SP	
14	AE	8F	D4	00014	CLRL	PUNC FLAG	4695
50	0200	AC	D0	0001D	CLRL	CURRENT_LEN	
18	AE	8F	3C	00017	MOVZWL	#512, STRING_PTR	4810
55	04	AC	D0	0001D	MOVL	STR_DESC, R0	4811
	0C	A0	D0	00021	MOVL	4(R0), STRING_PTR+4	
4C	AE	AC	D0	00026	MOVL	RET_LEN, R5	4813
50	AE	65	B4	0002A	CLRW	(R5)	
	30	11	D0	0002C	MOVL	#17, AFT_DATE_PTR	4827
54	04	AE	9E	00030	MOVAB	AFT_BUFF, AFT_DATE_PTR+4	4828
	24	AC	D0	00035	MOVL	SCB, R4	4833
	50	A4	9F	00039	PUSHAB	36(R4)	
	0C	AE	9F	0003C	PUSHAB	AFT_DATE_PTR	
	FDF0	AE	9F	0003F	PUSHAB	TEMP_LEN	
		CA	9F	00042	PUSHAB	DATE_FORMAT	
44	69	04	FB	00046	CALLS	#4, SYSSFAO	
48	AE	11	D0	00049	MOVL	#17, TEMP_PTR	4835
	1C	AE	9E	0004D	MOVAB	TEMP_BUFF, TEMP_PTR+4	4836
	015C	C4	9F	00052	PUSHAB	348(R4)	4841

			48	AE	9F	00056	PUSHAB	TEMP_PTR		
			0C	AE	9F	00059	PUSHAB	TEMP_LEN		
			FDF0	CA	9F	0005C	PUSHAB	DATE_FORMAT		
50	BE	48	BE	04	FB	00060	CALLS	#4, SYSSFAO		
				04	AE	29	CMPC3	TEMP_LEN, @TEMP_PTR+4, @AFT_DATE_PTR+4	4843	
				27	13	0006A	BEQL	1\$		
			24	A4	9F	0006C	PUSHAB	36(R4)	4851	
			18	AE	9F	0006F	PUSHAB	STRING_PTR		
			10	AE	9F	00072	PUSHAB	CURRENT_LEN		
			FE08	CA	9F	00075	PUSHAB	AFTER TIME FORMAT		
		69		04	FB	00079	CALLS	#4, SYSSFAO		
		65		08	AE	A0	ADDW2	CURRENT_LEN, (R5)	4853	
		18		08	AE	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	4854	
		14		65	3C	00085	MOVZWL	(R5), STRING_PTR	4855	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
			58	0154	C4	9E	MOVAB	340(R4), R8	4858	
			24	68	E9	00098	BLBC	(R8), 2\$		
				14	AE	9F	PUSHAB	STRING_PTR	4864	
				0C	AE	9F	PUSHAB	CURRENT_LEN		
				FE18	CA	9F	PUSHAB	BURST_FORMAT		
		69		03	FB	000A5	CALLS	#3, SYSSFAO		
		65		08	AE	A0	ADDW2	CURRENT_LEN, (R5)	4866	
		18		08	AE	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	4867	
		14		65	3C	000B1	MOVZWL	(R5), STRING_PTR	4868	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
				34	A4	B5	TSTW	52(R4)	4871	
				27	12	000C2	BNEQ	3\$		
				34	A4	9F	PUSHAB	52(R4)	4878	
				18	AE	9F	PUSHAB	STRING_PTR		
				10	AE	9F	PUSHAB	CURRENT_LEN		
				FE38	CA	9F	PUSHAB	CHARACTERISTICS_FORMAT		
		69		04	FB	000D1	CALLS	#4, SYSSFAO		
		65		08	AE	A0	ADDW2	CURRENT_LEN, (R5)	4880	
		18		08	AE	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	4881	
		14		65	3C	000DD	MOVZWL	(R5), STRING_PTR	4882	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
			57	0124	C4	9E	MOVAB	292(R4), R7	4887	
			67	02	E1	000F0	BBC	#2, (R7), 4\$		
				14	AE	9F	PUSHAB	STRING_PTR	4893	
				0C	AE	9F	PUSHAB	CURRENT_LEN		
				FE68	CA	9F	PUSHAB	FEED_FORMAT		
		69		03	FB	000FE	CALLS	#3, SYSSFAO		
		65		08	AE	A0	ADDW2	CURRENT_LEN, (R5)	4895	
		18		08	AE	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	4896	
		14		65	3C	0010A	MOVZWL	(R5), STRING_PTR	4897	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
			01	64	A4	D1	CMPL	100(R4), #1	4900	
				2B	15	0011C	BLEQ	5\$		
				64	A4	DD	PUSHL	100(R4)	4909	
				64	A4	7D	MOVQ	100(R4), -(SP)		
				20	AE	9F	PUSHAB	STRING_PTR		
				18	AE	9F	PUSHAB	CURRENT_LEN		
				FE58	CA	9F	PUSHAB	FILE_COPIES_FORMAT		
		69		06	FB	0012F	CALLS	#6, SYSSFAO		
		65		08	AE	A0	ADDW2	CURRENT_LEN, (R5)	4911	
		18		08	AE	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	4912	
		14		65	3C	0013B	MOVZWL	(R5), STRING_PTR	4913	

14	AE	00000200	8F	14	AE	C3	0013F	SUBL3	STRING_PTR, #512, STRING_PTR	
	24		68		01	E1	00149	BBC	#1, (R8), 6\$	4916
				14	AE	9F	0014D	PUSHAB	STRING_PTR	4922
				OC	AE	9F	00150	PUSHAB	CURRENT_LEN	
				FE78	CA	9F	00153	PUSHAB	FLAG_FORMAT	
			69		03	FB	00157	CALLS	#3, SYSSFAO	
			65	08	AE	A0	0015A	ADDW2	CURRENT_LEN, (R5)	4924
		18	AE	08	AE	C0	0015E	ADDL2	CURRENT_LEN, STRING_PTR+4	4925
		14	AE		65	3C	00163	MOVZWL	(R5), STRING_PTR	4926
14	AE	00000200	8F	14	AE	C3	00167	SUBL3	STRING_PTR, #512, STRING_PTR	
				0084	C4	9F	00171	PUSHAB	132(R4)	4932
				18	AE	9F	00175	PUSHAB	STRING_PTR	
				10	AE	9F	00178	PUSHAB	CURRENT_LEN	
				FE8C	CA	9F	0017B	PUSHAB	FORM_FORMAT	
			69		04	FB	0017F	CALLS	#4, SYSSFAO	
			65	08	AE	A0	00182	ADDW2	CURRENT_LEN, (R5)	4934
		18	AE	08	AE	C0	00186	ADDL2	CURRENT_LEN, STRING_PTR+4	4935
		14	AE		65	3C	0018B	MOVZWL	(R5), STRING_PTR	4936
14	AE	00000200	8F	14	AE	C3	0018F	SUBL3	STRING_PTR, #512, STRING_PTR	
		0200	8F		65	B1	00199	CMPW	(R5), #512	4938
				2D	1A	0019E	BGTRU	8\$		
	24		67		01	E1	001A0	BBC	#1, (R7), 7\$	4945
				14	AE	9F	001A4	PUSHAB	STRING_PTR	4951
				OC	AE	9F	001A7	PUSHAB	CURRENT_LEN	
				FE9C	CA	9F	001AA	PUSHAB	HEADER_FORMAT	
			69		03	FB	001AE	CALLS	#3, SYSSFAO	
			65	08	AE	A0	001B1	ADDW2	CURRENT_LEN, (R5)	4953
		18	AE	08	AE	C0	001B5	ADDL2	CURRENT_LEN, STRING_PTR+4	4954
		14	AE		65	3C	001BA	MOVZWL	(R5), STRING_PTR	4955
14	AE	00000200	8F	14	AE	C3	001BE	SUBL3	STRING_PTR, #512, STRING_PTR	
		0200	8F		65	B1	001C8	CMPW	(R5), #512	4958
					03	1B	001CD	BLEQU	9\$	
				051C	31	001CF	BRW	45\$		
			50	00A0	C4	DD	001D2	MOVL	160(R4), R0	4965
			01		50	D1	001D7	CMPL	R0, #1	
					2C	15	001DA	BLEQ	10\$	
					50	DD	001DC	PUSHL	R0	4974
				00A4	C4	DD	001DE	PUSHL	164(R4)	
					50	DD	001E2	PUSHL	R0	
				20	AE	9F	001E4	PUSHAB	STRING_PTR	
				18	AE	9F	001E7	PUSHAB	CURRENT_LEN	
				FE0	CA	9F	001EA	PUSHAB	JOB_COUNT_FORMAT	
			69		06	FB	001EE	CALLS	#6, SYSSFAO	
			65	08	AE	A0	001F1	ADDW2	CURRENT_LEN, (R5)	4976
		18	AE	08	AE	C0	001F5	ADDL2	CURRENT_LEN, STRING_PTR+4	4977
		14	AE		65	3C	001FA	MOVZWL	(R5), STRING_PTR	4978
14	AE	00000200	8F	14	AE	C3	001FE	SUBL3	STRING_PTR, #512, STRING_PTR	
		OC	AE	020E0000	8F	DD	00208	MOVL	#34471936, X_DESC	4987
				10	AE	D4	00210	CLRL	X_DESC+4	
				OC	AE	9F	00213	PUSHAB	LEN	4989
		04	AE		11	DD	00216	MOVL	#17, 4(SP)	4988
				04	AE	9F	0021A	PUSHAB	4(SP)	
					54	DD	0021D	PUSHL	R4	
		00000000G	00		03	FB	0021F	CALLS	#3, PSM\$READ_ITEM_DX	
				10	BE	DD	00226	PUSHL	@LEN+4	4995
				18	AE	9F	00229	PUSHAB	STRING_PTR	
				10	AE	9F	0022C	PUSHAB	CURRENT_LEN	

			FED4	CA	9F	0022F	PUSHAB	LENGTH FORMAT		
		69		04	FB	00233	CALLS	#4, SYSSFAO		
		65	08	AE	A0	00236	ADDW2	CURRENT_LEN, (R5)	4997	
	18	AE	08	AE	C0	0023A	ADDL2	CURRENT_LEN, STRING_PTR+4	4998	
	14	AE		65	3C	0023F	MOVZWL	(R5), STRING_PTR	4999	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
				00C0	C4	9F	PUSHAB	192(R4)	5008	
				18	AE	9F	PUSHAB	STRING_PTR		
				10	AE	9F	PUSHAB	CURRENT_LEN		
				FEEC	CA	9F	PUSHAB	LIBRARY_FORMAT		
		69		04	FB	0025B	CALLS	#4, SYSSFAO		
		65	08	AE	A0	0025E	ADDW2	CURRENT_LEN, (R5)	5010	
	18	AE	08	AE	C0	00262	ADDL2	CURRENT_LEN, STRING_PTR+4	5011	
	14	AE		65	3C	00267	MOVZWL	(R5), STRING_PTR	5012	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
	0200		8F	65	B1	00275	CMPW	(R5), #512	5014	
				03	1B	0027A	BLEQU	11\$		
				046F	31	0027C	BRW	45\$		
		52	0164	C4	D0	0027F	MOVL	356(R4), R2	5021	
				53	D4	00284	CLRL	R3		
				52	D5	00286	TSTL	R2		
				04	13	00288	BEQL	12\$		
				53	D6	0028A	INCL	R3		
				14	11	0028C	BRB	13\$		
				30	A4	D5	TSTL	48(R4)	5022	
				0F	12	00291	BNEQ	13\$		
				00BC	C4	D5	TSTL	188(R4)	5023	
				09	12	00297	BNEQ	13\$		
				0148	C4	D5	TSTL	328(R4)	5024	
				03	12	0029D	BNEQ	13\$		
				0196	31	0029F	BRW	21\$		
				14	AE	9F	PUSHAB	STRING_PTR	5031	
				0C	AE	9F	PUSHAB	CURRENT_LEN		
				FF00	CA	9F	PUSHAB	MARGIN_FORMAT		
		69		03	FB	002AC	CALLS	#3, SYSSFAO		
		65	08	AE	A0	002AF	ADDW2	CURRENT_LEN, (R5)	5033	
	18	AE	08	AE	C0	002B3	ADDL2	CURRENT_LEN, STRING_PTR+4	5034	
	14	AE		65	3C	002B8	MOVZWL	(R5), STRING_PTR	5035	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
			29	53	E9	002C6	BLBC	R3, 14\$	5037	
				52	DD	002C9	PUSHL	R2	5044	
				18	AE	9F	PUSHAB	STRING_PTR		
				10	AE	9F	PUSHAB	CURRENT_LEN		
				FF10	CA	9F	PUSHAB	TOP_FORMAT		
		69		04	FB	002D5	CALLS	#4, SYSSFAO		
		56		01	D0	002D8	MOVL	#1, PUNC_FLAG	5046	
		65	08	AE	A0	002DB	ADDW2	CURRENT_LEN, (R5)	5047	
	18	AE	08	AE	C0	002DF	ADDL2	CURRENT_LEN, STRING_PTR+4	5048	
	14	AE		65	3C	002E4	MOVZWL	(R5), STRING_PTR	5049	
14	AE	00000200	8F	14	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
				52	D4	002F2	CLRL	R2	5052	
				30	A4	D5	TSTL	48(R4)		
				2B	13	002F7	BEQL	15\$		
				52	D6	002F9	INCL	R2		
		26		56	E9	002FB	BLBC	PUNC_FLAG, 15\$	5053	
				00B0	CA	9F	PUSHAB	P.AFR	5061	
				18	AE	9F	PUSHAB	STRING_PTR		

		10	AE	9F	00305	PUSHAB	CURRENT_LEN		
			5A	DD	00308	PUSHL	R10		
	69		04	FB	0030A	CALLS	#4, SYSSFAO		
	65	08	AE	A0	0030D	ADDW2	CURRENT_LEN, (R5)	5063	
	18	08	AE	C0	00311	ADDL2	CURRENT_LEN, STRING_PTR+4	5064	
	14		AE	65	3C	00316	MOVZWL	(R5), STRING_PTR	5065
14	AE 00000200	8F	14	AE	C3	0031A	SUBL3	STRING_PTR, #512, STRING_PTR	
		2A		52	E9	00324	BLBC	R2, 16\$	5068
			30	A4	DD	00327	PUSHL	48(R4)	5076
			18	AE	9F	0032A	PUSHAB	STRING_PTR	
			10	AE	9F	0032D	PUSHAB	CURRENT_LEN	
		FF24		CA	9F	00330	PUSHAB	BOTTOM FORMAT	
	69		04	FB	00334	CALLS	#4, SYSSFAO		
	56		01	D0	00337	MOVL	#1, PUNC_FLAG	5078	
	65	08	AE	A0	0033A	ADDW2	CURRENT_LEN, (R5)	5079	
	18	08	AE	C0	0033E	ADDL2	CURRENT_LEN, STRING_PTR+4	5080	
	14		AE	65	3C	00343	MOVZWL	(R5), STRING_PTR	5081
14	AE 00000200	8F	14	AE	C3	00347	SUBL3	STRING_PTR, #512, STRING_PTR	
		52	00BC	C4	D0	00351	MOVL	188(R4), R2	5084
				53	D4	00356	CLRL	R3	
				52	D5	00358	TSTL	R2	
				2B	13	0035A	BEQL	17\$	
				53	D6	0035C	INCL	R3	
	26		56	E9	0035E	BLBC	PUNC_FLAG, 17\$	5085	
		00B2	CA	9F	00361	PUSHAB	P.AFC	5093	
		18	AE	9F	00365	PUSHAB	STRING_PTR		
		10	AE	9F	00368	PUSHAB	CURRENT_LEN		
			5A	DD	0036B	PUSHL	R10		
	69		04	FB	0036D	CALLS	#4, SYSSFAO		
	65	08	AE	A0	00370	ADDW2	CURRENT_LEN, (R5)	5094	
	18	08	AE	C0	00374	ADDL2	CURRENT_LEN, STRING_PTR+4	5095	
	14		AE	65	3C	00379	MOVZWL	(R5), STRING_PTR	5096
14	AE 00000200	8F	14	AE	C3	0037D	SUBL3	STRING_PTR, #512, STRING_PTR	
		29		53	E9	00387	BLBC	R3, 18\$	5099
				52	DD	0038A	PUSHL	R2	5106
			18	AE	9F	0038C	PUSHAB	STRING_PTR	
			10	AE	9F	0038F	PUSHAB	CURRENT_LEN	
		FF34	CA	9F	00392	PUSHAB	LEFT FORMAT		
	69		04	FB	00396	CALLS	#4, SYSSFAO		
	56		01	D0	00399	MOVL	#1, PUNC_FLAG	5108	
	65	08	AE	A0	0039C	ADDW2	CURRENT_LEN, (R5)	5109	
	18	08	AE	C0	003A0	ADDL2	CURRENT_LEN, STRING_PTR+4	5110	
	14		AE	65	3C	003A5	MOVZWL	(R5), STRING_PTR	5111
14	AE 00000200	8F	14	AE	C3	003A9	SUBL3	STRING_PTR, #512, STRING_PTR	
		52	0148	C4	D0	003B3	MOVL	328(R4), R2	5114
				53	D4	003B8	CLRL	R3	
				52	D5	003BA	TSTL	R2	
				2B	13	003BC	BEQL	19\$	
				53	D6	003BE	INCL	R3	
	26		56	E9	003C0	BLBC	PUNC_FLAG, 19\$	5115	
		00B4	CA	9F	003C3	PUSHAB	P.AFM	5123	
		18	AE	9F	003C7	PUSHAB	STRING_PTR		
		10	AE	9F	003CA	PUSHAB	CURRENT_LEN		
			5A	DD	003CD	PUSHL	R10		
	69		04	FB	003CF	CALLS	#4, SYSSFAO		
	65	08	AE	A0	003D2	ADDW2	CURRENT_LEN, (R5)	5125	
	18	AE	08	AE	C0	003D6	ADDL2	CURRENT_LEN, STRING_PTR+4	5126

14	AE	00000200	14	AE	8F	26	14	65	3C	003DB	MOVZWL	(R5), STRING_PTR	5127	
								AE	C3	003DF	SUBL3	STRING_PTR, #512, STRING_PTR		
								53	E9	003E9	BLBC	R3, 20\$	5130	
								52	DD	003EC	PUSHL	R2	5137	
							18	AE	9F	003EE	PUSHAB	STRING_PTR		
							10	AE	9F	003F1	PUSHAB	CURRENT_LEN		
							FF48	CA	9F	003F4	PUSHAB	RIGHT_FORMAT		
							69	04	FB	003F8	CALLS	#4, SYSSFAO		
							65	08	AE	A0	003FB	ADDW2	CURRENT_LEN, (R5)	5139
							18	08	AE	C0	003FF	ADDL2	CURRENT_LEN, STRING_PTR+4	5140
							14	65	3C	00404	MOVZWL	(R5), STRING_PTR	5141	
14	AE	00000200	14	AE	8F	8F	14	AE	C3	00408	SUBL3	STRING_PTR, #512, STRING_PTR		
							00B6	CA	9F	00412	PUSHAB	P.AFN	5149	
							18	AE	9F	00416	PUSHAB	STRING_PTR		
							10	AE	9F	00419	PUSHAB	CURRENT_LEN		
								5A	DD	0041C	PUSHL	R10		
							69	04	FB	0041E	CALLS	#4, SYSSFAO		
							65	08	AE	A0	00421	ADDW2	CURRENT_LEN, (R5)	5151
							18	08	AE	C0	00425	ADDL2	CURRENT_LEN, STRING_PTR+4	5152
							14	65	3C	0042A	MOVZWL	(R5), STRING_PTR	5153	
14	AE	00000200	14	AE	8F	8F	14	AE	C3	0042E	SUBL3	STRING_PTR, #512, STRING_PTR		
		0200						65	B1	00438	CMPW	(R5), #512	5156	
								69	1A	0043D	BGTRU	25\$		
							24	02	E0	0043F	BBS	#2, (R7), 22\$	5163	
								14	AE	9F	00443	PUSHAB	STRING_PTR	5169
							0C	AE	9F	00446	PUSHAB	CURRENT_LEN		
							FF58	CA	9F	00449	PUSHAB	NOFEED_FORMAT		
							69	03	FB	0044D	CALLS	#3, SYSSFAO		
							65	08	AE	A0	00450	ADDW2	CURRENT_LEN, (R5)	5171
							18	08	AE	C0	00454	ADDL2	CURRENT_LEN, STRING_PTR+4	5172
							14	65	3C	00459	MOVZWL	(R5), STRING_PTR	5173	
14	AE	00000200	14	AE	8F	8F	14	AE	C3	0045D	SUBL3	STRING_PTR, #512, STRING_PTR		
		0200						65	B1	00467	CMPW	(R5), #512	5176	
								3A	1A	0046C	BGTRU	25\$		
							74	A4	D5	0046E	TSTL	116(R4)	5183	
								06	12	00471	BNEQ	23\$		
							00B8	C4	D5	00473	TSTL	184(R4)	5184	
								2A	13	00477	BEQL	24\$		
							00B8	C4	DD	00479	PUSHL	184(R4)	5192	
							74	A4	DD	0047D	PUSHL	116(R4)		
							1C	AE	9F	00480	PUSHAB	STRING_PTR		
							14	AE	9F	00483	PUSHAB	CURRENT_LEN		
							90	AA	9F	00486	PUSHAB	PAGES_FORMAT		
							69	05	FB	00489	CALLS	#5, SYSSFAO		
							65	08	AE	A0	0048C	ADDW2	CURRENT_LEN, (R5)	5194
							18	08	AE	C0	00490	ADDL2	CURRENT_LEN, STRING_PTR+4	5195
							14	65	3C	00495	MOVZWL	(R5), STRING_PTR	5196	
14	AE	00000200	14	AE	8F	8F	14	AE	C3	00499	SUBL3	STRING_PTR, #512, STRING_PTR		
		0200						65	B1	004A3	CMPW	(R5), #512	5199	
								78	1A	004A8	BGTRU	28\$		
							00E4	C4	B5	004AA	TSTW	228(R4)	5206	
								2A	12	004AE	BNEQ	26\$		
							00EC	C4	B5	004B0	TSTW	236(R4)	5207	
								24	12	004B4	BNEQ	26\$		
							00F4	C4	B5	004B6	TSTW	244(R4)	5208	
								1E	12	004BA	BNEQ	26\$		
							00FC	C4	B5	004BC	TSTW	252(R4)	5209	

			0104	18	12	004C0	BNEQ	26\$		
				C4	B5	004C2	TSTW	260(R4)	5210	
				12	12	004C6	BNEQ	26\$		
			010C	C4	B5	004C8	TSTW	268(R4)	5211	
				0C	12	004CC	BNEQ	26\$		
			0114	C4	B5	004CE	TSTW	276(R4)	5212	
				06	12	004D2	BNEQ	26\$		
			011C	C4	B5	004D4	TSTW	284(R4)	5213	
				43	13	004D8	BEQL	27\$		
			011C	C4	9F	004DA	PUSHAB	284(R4)	5228	
			0114	C4	9F	004DE	PUSHAB	276(R4)		
			010C	C4	9F	004E2	PUSHAB	268(R4)		
			0104	C4	9F	004E6	PUSHAB	260(R4)		
			00FC	C4	9F	004EA	PUSHAB	252(R4)		
			00F4	C4	9F	004EE	PUSHAB	244(R4)		
			00EC	C4	9F	004F2	PUSHAB	236(R4)		
			00E4	C4	9F	004F6	PUSHAB	228(R4)		
			34	AE	9F	004FA	PUSHAB	STRING_PTR		
			2C	AE	9F	004FD	PUSHAB	CURRENT_LEN		
			E0	AA	9F	00500	PUSHAB	PARAMETER_FORMAT		
		69		0B	FB	00503	CALLS	#11, SYSSFAO		
		65	08	AE	A0	00506	ADDW2	CURRENT_LEN, (R5)	5230	
		18	08	AE	C0	0050A	ADDL2	CURRENT_LEN, STRING_PTR+4	5231	
		14	65	3C	0050F	MOVZWL	(R5), STRING_PTR	5232		
14	AE	00000200	14	AE	C3	00513	SUBL3	STRING_PTR, #512, STRING_PTR		
		0200	65	B1	0051D	CMPL	(R5), #512	5235		
			5E	1A	00522	BGTRU	31\$			
		23	03	E1	00524	BBC	#3, (R7), 29\$	5242		
		67	14	AE	9F	00528	PUSHAB	STRING_PTR	5248	
			0C	AE	9F	0052B	PUSHAB	CURRENT_LEN		
			F4	AA	9F	0052E	PUSHAB	PASSALL_FORMAT		
				03	FB	00531	CALLS	#3, SYSSFAO		
		69	08	AE	A0	00534	ADDW2	CURRENT_LEN, (R5)	5250	
		65	08	AE	C0	00538	ADDL2	CURRENT_LEN, STRING_PTR+4	5251	
		18	65	3C	0053D	MOVZWL	(R5), STRING_PTR	5252		
14	AE	00000200	14	AE	C3	00541	SUBL3	STRING_PTR, #512, STRING_PTR		
		0200	65	B1	0054B	CMPL	(R5), #512	5255		
			30	1A	00550	BGTRU	31\$			
			6C	A4	B5	00552	TSTW	108(R4)	5262	
				26	13	00555	BEQL	30\$		
			6C	A4	9F	00557	PUSHAB	108(R4)	5269	
			18	AE	9F	0055A	PUSHAB	STRING_PTR		
			10	AE	9F	0055D	PUSHAB	CURRENT_LEN		
			1C	AA	9F	00560	PUSHAB	SETUP_FILE_FORMAT		
				04	FB	00563	CALLS	#4, SYSSFAO		
		69	08	AE	A0	00566	ADDW2	CURRENT_LEN, (R5)	5271	
		65	08	AE	C0	0056A	ADDL2	CURRENT_LEN, STRING_PTR+4	5272	
		18	65	3C	0056F	MOVZWL	(R5), STRING_PTR	5273		
14	AE	00000200	14	AE	C3	00573	SUBL3	STRING_PTR, #512, STRING_PTR		
		0200	65	B1	0057D	CMPL	(R5), #512	5276		
			5E	1A	00582	BGTRU	34\$			
			7C	A4	B5	00584	TSTW	124(R4)	5283	
				26	13	00587	BEQL	32\$		
			7C	A4	9F	00589	PUSHAB	124(R4)	5290	
			18	AE	9F	0058C	PUSHAB	STRING_PTR		
			10	AE	9F	0058F	PUSHAB	CURRENT_LEN		
			38	AA	9F	00592	PUSHAB	SETUP_FORM_FORMAT		

14	AE	00000200	69	04	FB	00595	CALLS	#4, SYSSFAO	5292
			65	08	AE	A0 00598	ADDW2	CURRENT_LEN, (R5)	5293
			18	08	AE	C0 0059C	ADDL2	CURRENT_LEN, STRING_PTR+4	5294
			14		65	3C 005A1	MOVZWL	(R5), STRING_PTR	5297
			8F	14	AE	C3 005A5	SUBL3	STRING_PTR, #512, STRING_PTR	5304
				00DC	C4	B5 005AF	TSTW	220(R4)	
					28	13 005B3	BEQL	33\$	
				00DC	C4	9F 005B5	PUSHAB	220(R4)	
				18	AE	9F 005B9	PUSHAB	STRING_PTR	
				10	AE	9F 005BC	PUSHAB	CURRENT_LEN	
				FF74	CA	9F 005BF	PUSHAB	SETUP PAGE FORMAT	
					04	FB	005C3	CALLS	#4, SYSSFAO
			69	08	AE	A0 005C6	ADDW2	CURRENT_LEN, (R5)	5306
			18	08	AE	C0 005CA	ADDL2	CURRENT_LEN, STRING_PTR+4	5307
			14		65	3C 005CF	MOVZWL	(R5), STRING_PTR	5308
			8F	14	AE	C3 005D3	SUBL3	STRING_PTR, #512, STRING_PTR	5311
			8F		65	B1 005DD	CMPW	(R5), #512	
					59	1A 005E2	BGTRU	37\$	
					05	E1 005E4	BBC	#5, (R7), 35\$	5318
				14	AE	9F 005E8	PUSHAB	STRING_PTR	5324
				0C	AE	9F 005EB	PUSHAB	CURRENT_LEN	
				4C	AA	9F 005EE	PUSHAB	SHEET FORMAT	
					03	FB	005F1	CALLS	#3, SYSSFAO
			69	08	AE	A0 005F4	ADDW2	CURRENT_LEN, (R5)	5326
			18	08	AE	C0 005F8	ADDL2	CURRENT_LEN, STRING_PTR+4	5327
			14		65	3C 005FD	MOVZWL	(R5), STRING_PTR	5328
			8F	14	AE	C3 00601	SUBL3	STRING_PTR, #512, STRING_PTR	5331
			8F		65	B1 0060B	CMPW	(R5), #512	
					59	1A 00610	BGTRU	39\$	
					67	E9 00612	BLBC	(R7), 36\$	5338
				14	AE	9F 00615	PUSHAB	STRING_PTR	5344
				0C	AE	9F 00618	PUSHAB	CURRENT_LEN	
				5C	AA	9F 0061B	PUSHAB	SPACE FORMAT	
					03	FB	0061E	CALLS	#3, SYSSFAO
			69	08	AE	A0 00621	ADDW2	CURRENT_LEN, (R5)	5346
			18	08	AE	C0 00625	ADDL2	CURRENT_LEN, STRING_PTR+4	5347
			14		65	3C 0062A	MOVZWL	(R5), STRING_PTR	5348
			8F	14	AE	C3 0062E	SUBL3	STRING_PTR, #512, STRING_PTR	5351
			8F		65	B1 00638	CMPW	(R5), #512	
					5B	1A 0063D	BGTRU	41\$	
					02	E1 0063F	BBC	#2, (R8), 38\$	5358
				14	AE	9F 00643	PUSHAB	STRING_PTR	5364
				0C	AE	9F 00646	PUSHAB	CURRENT_LEN	
				70	AA	9F 00649	PUSHAB	TRAILER FORMAT	
					03	FB	0064C	CALLS	#3, SYSSFAO
			69	08	AE	A0 0064F	ADDW2	CURRENT_LEN, (R5)	5366
			18	08	AE	C0 00653	ADDL2	CURRENT_LEN, STRING_PTR+4	5367
			14		65	3C 00658	MOVZWL	(R5), STRING_PTR	5368
			8F	14	AE	C3 0065C	SUBL3	STRING_PTR, #512, STRING_PTR	5371
			8F		65	B1 00666	CMPW	(R5), #512	
					5C	1A 0066B	BGTRU	42\$	
					06	E1 0066D	BBC	#6, (R7), 40\$	5378
				14	AE	9F 00671	PUSHAB	STRING_PTR	5384
				0C	AE	9F 00674	PUSHAB	CURRENT_LEN	
				0084	CA	9F 00677	PUSHAB	TRUNCATE FORMAT	
					03	FB	0067B	CALLS	#3, SYSSFAO
			69	08	AE	A0 0067E	ADDW2	CURRENT_LEN, (R5)	5386
			65						

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifiers associ

K 8
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 149
(34)

18	AE	08	AE	C0	00682	ADDL2	CURRENT_LEN, STRING_PTR+4	5387	
14	AE		65	3C	00687	MOVZWL	(R5), STRING_PTR	5388	
14	AE	00000200	8F	14	AE	C3	00688	SUBL3	STRING_PTR, #512, STRING_PTR
0200	8F		65	B1	00695	40\$:	CMPW	(R5), #512	5391
			52	1A	0069A	41\$:	BGTRU	45\$	5402
		008C	C4	DD	0069C		PUSHL	140(R4)	
		18	AE	9F	006A0		PUSHAB	STRING_PTR	
		10	AE	9F	006A3		PUSHAB	CURRENT_LEN	
		0098	CA	9F	006A6		PUSHAB	WIDTH FORMAT	
	69		04	FB	006AA		CALLS	#4, SYS\$FAO	
	65	08	AE	A0	006AD		ADDW2	CURRENT_LEN, (R5)	5404
	18	08	AE	C0	006B1		ADDL2	CURRENT_LEN, STRING_PTR+4	5405
	14		65	3C	006B6		MOVZWL	(R5), STRING_PTR	5406
14	AE	00000200	8F	14	AE	C3	006BA	SUBL3	STRING_PTR, #512, STRING_PTR
0200	8F		65	B1	006C4		CMPW	(R5), #512	5408
			23	1A	006C9	42\$:	BGTRU	45\$	
			67	95	006CB		TSTB	(R7)	5415
			11	18	006CD		BGEQ	43\$	
		14	AE	9F	006CF		PUSHAB	STRING_PTR	5421
		0C	AE	9F	006D2		PUSHAB	CURRENT_LEN	
		00A8	CA	9F	006D5		PUSHAB	WRAP FORMAT	
	69		03	FB	006D9		CALLS	#3, SYS\$FAO	
	65	08	AE	A0	006DC		ADDW2	CURRENT_LEN, (R5)	5423
	12		65	B1	006E0	43\$:	CMPW	(R5), #T8	5428
			02	1A	006E3		BGTRU	44\$	
			65	B4	006E5		CLRW	(R5)	5430
0200	8F		65	B1	006E7	44\$:	CMPW	(R5), #512	5434
			05	1B	006EC		BLEQU	46\$	
	65	0200	8F	B0	006EE	45\$:	MOVW	#512, (R5)	5437
			04	006F3	46\$:	RET			5441

; Routine Size: 1780 bytes, Routine Base: CODE + 2070

```
4529 5442 1 %sbttl 'GET_QUEUE_QUALIFIERS - Gets the qualifiers pertaining to queues'
4530 5443 1 ++
4531 5444 1 Functional Description:
4532 5445 1 This routine returns a string containing the all relevant file qualifier
4533 5446 1 information.
4534 5447 1
4535 5448 1 Formal Parameters:
4536 5449 1 SCB - Address of the SCB
4537 5450 1 STR_DESC - Desc of String to Return
4538 5451 1 RET_LEN - Return length of Desc.
4539 5452 1
4540 5453 1 Implicit Inputs:
4541 5454 1 none
4542 5455 1
4543 5456 1 Implicit Outputs:
4544 5457 1 none
4545 5458 1
4546 5459 1 Returned Value:
4547 5460 1 none
4548 5461 1
4549 5462 1 Side Effects:
4550 5463 1 none
4551 5464 1 --
4552 5465 1 ROUTINE GET_QUEUE_QUALIFIERS (
4553 5466 1 SCB : REF $BLOCK, ! SCB
4554 5467 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
4555 5468 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
4556 5469 1 ) : NOVALUE =
4557 5470 2 BEGIN
4558 5471 2 BIND
4559 P 5472 2 BEGIN FORMAT = $DESCRIPTOR (
4560 5473 2 'Queue Qualifiers:'),
4561 5474 2
4562 P 5475 2 RESET FORMAT = $DESCRIPTOR (
4563 P 5476 2 'RESET=',
4564 P 5477 2 '!AS', ! - Reset Module
4565 5478 2 ' '),
4566 5479 2
4567 P 5480 2 JOB_RESET_MODULE FORMAT = $DESCRIPTOR (
4568 5481 2 '/SEPARATE=T'), ! - separation
4569 5482 2
4570 P 5483 2 INSERTION FORMAT = $DESCRIPTOR (
4571 5484 2 '!AC'T; ! - separation flags
4572 5485 2
4573 5486 2
4574 5487 2 LITERAL
4575 5488 2 K_MAX_BUFFER_SIZE = 512;
4576 5489 2
4577 5490 2 LOCAL
4578 5491 2 INSERT_FLAG : INITIAL (0),
4579 5492 2 CURRENT_LEN : INITIAL (0),
4580 5493 2 STRING_PTR : VECTOR [2]; ! Pointer to current string
4581 5494 2
4582 5495 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
4583 5496 2 STRING_PTR[ADDR] = .STR_DESC[ADDR]; ! init address
4584 5497 2
4585 5498 2 RET_LEN[0] = 0;
```

```
4586      5499  2
4587      5500  2 $FAO ( BEGIN FORMAT,
4588      5501  2      CURRENT_LEN,           ! return length
4589      5502  2      STRING_PTR[0],         ! address of string
4590      5503  2      );
4591      5504  2
4592      5505  2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4593      5506  2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4594      5507  2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4595      5508  2
4596      5509  2
4597      5510  2 IF .SEPARATE_FLAG_ (JOB BURST) OR
4598      5511  2     .SEPARATE_FLAG_ (JOB_FLAG) OR
4599      5512  2     .SEPARATE_FLAG_ (JOB_TRAILER) OR
4600      5513  2     .SCB_SIZE_ (JOB_RESET_MODULES)
4601      5514  2 THEN
4602      5515  2 BEGIN
4603      5516  2 $FAO ( JOB_RESET_MODULE_FORMAT, ! /SEPARATE
4604      5517  2      CURRENT_LEN,           ! return length
4605      5518  2      STRING_PTR[0]         ! address of string
4606      5519  2      );
4607      5520  2
4608      5521  2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4609      5522  2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4610      5523  2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4611      5524  2
4612      5525  2 IF .SEPARATE_FLAG_ (JOB_BURST)
4613      5526  2 THEN
4614      5527  2 BEGIN
4615      5528  2 $FAO (
4616      5529  2     INSERTION FORMAT,
4617      5530  2     CURRENT_LEN,           ! return length
4618      5531  2     STRING_PTR[0],         ! address of string
4619      5532  2     UPLIT BYTE ('BURST')
4620      5533  2     );
4621      5534  2
4622      5535  2 INSERT_FLAG = 1;
4623      5536  2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4624      5537  2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4625      5538  2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4626      5539  2 END;
4627      5540  2
4628      5541  2 IF .SEPARATE_FLAG_ (JOB_FLAG) AND
4629      5542  2     .INSERT_FLAG
4630      5543  2 THEN
4631      5544  2 BEGIN
4632      5545  2 $FAO (
4633      5546  2     INSERTION FORMAT,
4634      5547  2     CURRENT_LEN,           ! return length
4635      5548  2     STRING_PTR[0],         ! address of string
4636      5549  2     UPLIT BYTE ('.', '.')
4637      5550  2     );
4638      5551  2
4639      5552  2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4640      5553  2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4641      5554  2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4642      5555  2 END;
```

```
4643 5556 3
4644 5557 3 IF .SEPARATE_FLAG_ (JOB_FLAG)
4645 5558 3 THEN
4646 5559 4 BEGIN
4647 P 5560 4 $FAO (
4648 P 5561 4 INSERTION FORMAT,
4649 P 5562 4 CURRENT_LEN, ! return length
4650 P 5563 4 STRING_PTR[0], ! address of string
4651 P 5564 4 UPLIT BYTE (%ASCIC'FLAG')
4652 5565 4 );
4653 5566 4
4654 5567 4 INSERT_FLAG = 1;
4655 5568 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4656 5569 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4657 5570 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4658 5571 4 END;
4659 5572 3
4660 5573 3 IF .SCB_SIZE_ (JOB_RESET_MODULES) AND
4661 5574 3 .INSERT_FLAG
4662 5575 3 THEN
4663 5576 4 BEGIN
4664 P 5577 4 $FAO (
4665 P 5578 4 INSERTION FORMAT,
4666 P 5579 4 CURRENT_LEN, ! return length
4667 P 5580 4 STRING_PTR[0], ! address of string
4668 P 5581 4 UPLIT BYTE (%ASCIC',')
4669 5582 4 );
4670 5583 4
4671 5584 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4672 5585 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4673 5586 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4674 5587 4 END;
4675 5588 3
4676 5589 3 IF .SCB_SIZE_ (JOB_RESET_MODULES)
4677 5590 3 THEN
4678 5591 4 BEGIN
4679 P 5592 4 $FAO (
4680 P 5593 4 RESET FORMAT,
4681 P 5594 4 CURRENT_LEN, ! return length
4682 P 5595 4 STRING_PTR[0], ! address of string
4683 P 5596 4 SCB[PSM$Q_JOB_RESET_MODULES]
4684 5597 4 );
4685 5598 4
4686 5599 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4687 5600 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4688 5601 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4689 5602 4 END;
4690 5603 3
4691 5604 3 IF .SEPARATE_FLAG_ (JOB_TRAILER) AND
4692 5605 3 .INSERT_FLAG
4693 5606 3 THEN
4694 5607 4 BEGIN
4695 P 5608 4 $FAO (
4696 P 5609 4 INSERTION FORMAT,
4697 P 5610 4 CURRENT_LEN, ! return length
4698 P 5611 4 STRING_PTR[0], ! address of string
4699 P 5612 4 UPLIT BYTE (%ASCIC',')
```



```
4700 5613 4 );
4701 5614 4
4702 5615 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4703 5616 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4704 5617 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4705 5618 4 END;
4706 5619 4
4707 5620 4 IF .SEPARATE_FLAG_ (JOB_TRAILER)
4708 5621 4 THEN
4709 5622 4 BEGIN
4710 P 5623 4 $FAO (
4711 P P 5624 4 INSERTION FORMAT,
4712 P P 5625 4 CURRENT_LEN, ; return length
4713 P P 5626 4 STRING_PTR[0], ; address of string
4714 P 5627 4 UPLIT BYTE ('TRAILER')
4715 5628 4 );
4716 5629 4
4717 5630 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4718 5631 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4719 5632 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4720 5633 4 END;
4721 5634 4
4722 P 5635 4 $FAO (
4723 P P 5636 4 INSERTION FORMAT,
4724 P P 5637 4 CURRENT_LEN, ; return length
4725 P P 5638 4 STRING_PTR[0], ; address of string
4726 P 5639 4 UPLIT BYTE ('')
4727 5640 4 );
4728 5641 4
4729 5642 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4730 5643 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4731 5644 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4732 5645 4 END;
4733 5646 4
4734 5647 4 ! Don't print anything if no flags were set
4735 5648 4
4736 5649 4 IF .RET_LEN[0] LEQ 18
4737 5650 4 THEN
4738 5651 4 RET_LEN[0] = 0;
4739 5652 4
4740 5653 4 ! Length returned must be less than max string size
4741 5654 4 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4742 5655 4 THEN
4743 5656 4 BEGIN
4744 5657 4 RET_LEN[0] = 512;
4745 5658 4 RETURN;
4746 5659 4 END;
4747 5660 4
4748 5661 1 END;
```

```
72 65 69 66 69 60 61 75 51 20 65 75 65 75 51 02764 P.AFP: .ASCII \Queue Qualifiers:\
3A 73 02773
02775
00000011 02778 P.AFO: .BLKB 3
00000000 0277C .LONG 17
ADDRESS P.AFP
```

```
22 3D 54 45 53 45 52 02780 P.AFR: .ASCII \RESET='\
53 41 21 02787 .ASCII \!AS\
22 0278A .ASCII \'\
0278B .BLKB 1
0000000B 0278C P.AFQ: .LONG 11
00000000 02790 .ADDRESS P.AFR
28 3D 45 54 41 52 41 50 45 53 2F 20 02794 P.AFT: .ASCII \ /SEPARATE=(\
0000000C 027A0 P.AFS: .LONG 12
00000000 027A4 .ADDRESS P.AFT
43 41 21 027A8 P.AFV: .ASCII \!AC\
027AB .BLKB 1
00000003 027AC P.AFU: .LONG 3
00000000 027B0 .ADDRESS P.AFV
54 53 52 55 42 05 027B4 P.AFW: .ASCII <5>\BURST\
2C 01 027BA P.AFX: .ASCII <1>\,\
47 41 4C 46 04 027BC P.AFY: .ASCII <4>\FLAG\
2C 01 027C1 P.AFZ: .ASCII <1>\,\
2C 01 027C3 P.AGA: .ASCII <1>\,\
52 45 4C 49 41 52 54 07 027C5 P.AGB: .ASCII <7>\TRAILER\
29 01 027CD P.AGC: .ASCII <1>\)\
```

```
BEGIN_FORMAT= P.AFO
RESET_FORMAT= P.AFQ
JOB_RESET_MODULE_FORMAT=
INSERTION_FORMAT= P.AFS
P.AFU
```

```
00FC 00000 GET_QUEUE_QUALIFIERS:
57 D8 AF 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7 5465
56 00000000G 00 9E 00006 MOVAB INSERTION_FORMAT, R7
5E 08 C2 0000D MOVAB SYSSFAO, R6
55 D4 00010 SUBL2 #8, SP
7E D4 00012 CLRL INSERT_FLAG 5470
04 AE 0200 8F 3C 00014 CLRL CURRENT_LEN
50 08 AC D0 0001A MOVZWL #512, STRING_PTR 5495
08 AE 04 A0 D0 0001E MOVL STR_DESC, R0 5496
52 0C AC D0 00023 MOVL 4(R0), STRING_PTR+4
62 B4 00027 MOVL RET_LEN, R2 5498
04 AE 9F 00029 CLRW (R2)
04 AE 9F 0002C PUSHAB STRING_PTR 5503
CC A7 9F 0002F PUSHAB CURRENT_LEN
66 03 FB 00032 PUSHAB BEGIN_FORMAT
62 6E A0 00035 CALLS #3, SYSSFAO
08 AE 6E C0 00038 ADDW2 CURRENT_LEN, (R2) 5505
04 AE 62 3C 0003C ADDL2 CURRENT_LEN, STRING_PTR+4 5506
04 AE 8F 04 AE C3 00040 MOVZWL (R2), STRING_PTR 5507
53 04 AC D0 0004A SUBL3 STRING_PTR, #512, STRING_PTR
54 0154 C3 9E 0004E MOVL SCB, R3 5510
64 05 E0 00053 MOVAB 340(R3), R4
64 04 E0 00057 BBS #5, (R4), 1$
08 01 A4 E8 0005B BBS #4, (R4), 1$
03 00B0 C3 E8 0005F BLBS 1(R4), 1$
04 0168 31 00064 BRW 9$ 5511
04 AE 9F 00067 1$: PUSHAB STRING_PTR 5512
04 AE 9F 0006A PUSHAB CURRENT_LEN 5513
```

			F4	A7	9F	0006D	PUSHAB	JOB_RESET_MODULE_FORMAT		
		66		03	FB	00070	CALLS	#3, SYSSFAO		
		62		6E	A0	00073	ADDW2	CURRENT_LEN, (R2)	5521	
	08	AE		6E	C0	00076	ADDL2	CURRENT_LEN, STRING_PTR+4	5522	
	04	AE		62	3C	0007A	MOVZWL	(R2), STRING_PTR	5523	
04	AE	00000200		04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
	26			05	E1	00088	BBC	#5, (R4), 2\$	5525	
				08	A7	9F	PUSHAB	P.AFW	5533	
				08	AE	9F	PUSHAB	STRING_PTR		
				08	AE	9F	PUSHAB	CURRENT_LEN		
				57	DD	00095	PUSHL	R7		
		66		04	FB	00097	CALLS	#4, SYSSFAO		
		55		01	D0	0009A	MOVL	#1, INSERT_FLAG	5535	
		62		6E	A0	0009D	ADDW2	CURRENT_LEN, (R2)	5536	
	08	AE		6E	C0	000A0	ADDL2	CURRENT_LEN, STRING_PTR+4	5537	
	04	AE		62	3C	000A4	MOVZWL	(R2), STRING_PTR	5538	
04	AE	00000200		04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
	50			04	E1	000B2	BBC	#4, (R4), 4\$	5541	
				55	E9	000B6	BLBC	INSERT_FLAG, 3\$	5542	
				0E	A7	9F	PUSHAB	P.AFX	5550	
				08	AE	9F	PUSHAB	STRING_PTR		
				08	AE	9F	PUSHAB	CURRENT_LEN		
				57	DD	000C2	PUSHL	R7		
		66		04	FB	000C4	CALLS	#4, SYSSFAO		
		62		6E	A0	000C7	ADDW2	CURRENT_LEN, (R2)	5552	
	08	AE		6E	C0	000CA	ADDL2	CURRENT_LEN, STRING_PTR+4	5553	
	04	AE		62	3C	000CE	MOVZWL	(R2), STRING_PTR	5554	
04	AE	00000200		04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
	26			04	E1	000DC	BBC	#4, (R4), 4\$	5557	
				10	A7	9F	PUSHAB	P.AFY	5565	
				08	AE	9F	PUSHAB	STRING_PTR		
				08	AE	9F	PUSHAB	CURRENT_LEN		
				57	DD	000E9	PUSHL	R7		
		66		04	FB	000EB	CALLS	#4, SYSSFAO		
		55		01	D0	000EE	MOVL	#1, INSERT_FLAG	5567	
		62		6E	A0	000F1	ADDW2	CURRENT_LEN, (R2)	5568	
	08	AE		6E	C0	000F4	ADDL2	CURRENT_LEN, STRING_PTR+4	5569	
	04	AE		62	3C	000F8	MOVZWL	(R2), STRING_PTR	5570	
04	AE	00000200		04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
			00B0	C3	E9	00106	BLBC	176(R3), 6\$	5573	
				55	E9	0010B	BLBC	INSERT_FLAG, 5\$	5574	
				15	A7	9F	PUSHAB	P.AFZ	5582	
				08	AE	9F	PUSHAB	STRING_PTR		
				08	AE	9F	PUSHAB	CURRENT_LEN		
				57	DD	00117	PUSHL	R7		
		66		04	FB	00119	CALLS	#4, SYSSFAO		
		62		6E	A0	0011C	ADDW2	CURRENT_LEN, (R2)	5584	
	08	AE		6E	C0	0011F	ADDL2	CURRENT_LEN, STRING_PTR+4	5585	
	04	AE		62	3C	00123	MOVZWL	(R2), STRING_PTR	5586	
04	AE	00000200		04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR		
			00B0	C3	E9	00131	BLBC	176(R3), 6\$	5589	
			00B0	C3	9F	00136	PUSHAB	176(R3)	5597	
				08	AE	9F	PUSHAB	STRING_PTR		
				08	AE	9F	PUSHAB	CURRENT_LEN		
			E0	A7	9F	00140	PUSHAB	RESET_FORMAT		
		66		04	FB	00143	CALLS	#4, SYSSFAO		
		62		6E	A0	00146	ADDW2	CURRENT_LEN, (R2)	5599	

```
Print Symbiont -- separation routines
GET_QUEUE_QUALIFIERS - Gets the quali
```

GET_QUEUE_QUALIFIERS - Gets the qualifiers pert

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 BLISS-32 V4.0-742

[PRTSMB.SRC]SEPARATE.B32:2

Page 156

(35)

SEP
V04

		08	AE	6E	C0	00149	ADDL2	CURRENT_LEN, STRING_PTR+4	:	5600
		04	AE	62	3C	0014D	MOVZWL	(R2), STRING_PTR	:	5601
04	AE	00000200	8F	AE	C3	00151	SUBL3	STRING_PTR, #512, STRING_PTR	:	
			4D	01	A4	E9	BLBC	1(R4), -8\$:	5604
			23		55	E9	BLBC	INSERT_FLAG, 7\$:	5605
				17	A7	9F	PUSHAB	P.AGA	:	5613
				08	AE	9F	PUSHAB	STRING_PTR	:	
				08	AE	9F	PUSHAB	CURRENT_LEN	:	
					57	DD	PUSHL	R7	:	
		66			04	FB	CALLS	#4, SYSSFAO	:	
		62			6E	A0	ADDW2	CURRENT_LEN, (R2)	:	5615
		08	AE		6E	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	:	5616
		04	AE		62	3C	MOVZWL	(R2), STRING_PTR	:	5617
04	AE	00000200	8F	04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR	:	
			23	01	A4	E9	BLBC	1(R4), -8\$:	5620
				19	A7	9F	PUSHAB	P.AGB	:	5628
				08	AE	9F	PUSHAB	STRING_PTR	:	
				08	AE	9F	PUSHAB	CURRENT_LEN	:	
					57	DD	PUSHL	R7	:	
		66			04	FB	CALLS	#4, SYSSFAO	:	
		62			6E	A0	ADDW2	CURRENT_LEN, (R2)	:	5630
		08	AE		6E	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	:	5631
		04	AE		62	3C	MOVZWL	(R2), STRING_PTR	:	5632
04	AE	00000200	8F	04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR	:	
				21	A7	9F	PUSHAB	P.AGC	:	5640
				08	AE	9F	PUSHAB	STRING_PTR	:	
				08	AE	9F	PUSHAB	CURRENT_LEN	:	
					57	DD	PUSHL	R7	:	
		66			04	FB	CALLS	#4, SYSSFAO	:	
		62			6E	A0	ADDW2	CURRENT_LEN, (R2)	:	5642
		08	AE		6E	C0	ADDL2	CURRENT_LEN, STRING_PTR+4	:	5643
		04	AE		62	3C	MOVZWL	(R2), STRING_PTR	:	5644
04	AE	00000200	8F	04	AE	C3	SUBL3	STRING_PTR, #512, STRING_PTR	:	
			12		62	B1	CMPW	(R2), #18	:	5649
					02	1A	BGTRU	10\$:	
					62	B4	CLRW	(R2)	:	5651
		0200	8F		62	B1	CMPW	(R2), #512	:	5654
					05	1B	BLEQU	11\$:	
			62	0200	8F	B0	MOVW	#512, (R2)	:	5657
					04	001E2	RET		:	5661

; Routine Size: 483 bytes, Routine Base: CODE + 27CF


```
4750 5662 1 %sbttl 'GET_FORM_QUALIFIERS - Get the qualifiers which pertain to forms'
4751 5663 1 ++
4752 5664 1 Functional Description:
4753 5665 1 This routine returns a string containing the all relevant file qualifier
4754 5666 1 information.
4755 5667 1
4756 5668 1 Formal Parameters:
4757 5669 1 SCB - Address of the SCB
4758 5670 1 STR_DESC - Desc of String to Return
4759 5671 1 RET_LEN - Return length of Desc.
4760 5672 1
4761 5673 1 Implicit Inputs:
4762 5674 1 none
4763 5675 1
4764 5676 1 Implicit Outputs:
4765 5677 1 none
4766 5678 1
4767 5679 1 Returned Value:
4768 5680 1 none
4769 5681 1
4770 5682 1 Side Effects:
4771 5683 1 none
4772 5684 1 --
4773 5685 1 ROUTINE GET_FORM_QUALIFIERS (
4774 5686 1 SCB : REF $BBLOCK, ! SCB
4775 5687 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
4776 5688 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
4777 5689 1 ) : NOVALUE =
4778 5690 2 BEGIN
4779 5691 2 BIND
4780 P 5692 2 BEGIN FORMAT = $DESCRIPTOR(
4781 5693 2 'Form Qualifiers:');
4782 5694 2
4783 5695 2 LITERAL
4784 5696 2 K_MAX_BUFFER_SIZE = 512;
4785 5697 2
4786 5698 2 LOCAL
4787 5699 2 CURRENT_LEN : INITIAL (0),
4788 5700 2 STRING_PTR : VECTOR [2]; ! Pointer to current string
4789 5701 2
4790 5702 2
4791 5703 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
4792 5704 2 STRING_PTR[ADDR] = .STR_DESC[ADDR]; ! init address
4793 5705 2
4794 5706 2 RET_LEN[0] = 0;
4795 5707 2
4796 P 5708 2 $FAO ( BEGIN FORMAT,
4797 P 5709 2 CURRENT_LEN, ! return length
4798 P 5710 2 STRING_PTR[0], ! address of string
4799 5711 2 );
4800 5712 2
4801 5713 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4802 5714 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4803 5715 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4804 5716 2
4805 5717 2
4806 5718 2
```

```

4807      5719      2      ! Don't print anything if no flags were set
4808      5720      2
4809      5721      2      IF .RET_LEN[0] LEQ 18
4810      5722      2      THEN
4811      5723      2          RET_LEN[0] = 0;
4812      5724      2
4813      5725      2      ! Length returned must be less than max string size
4814      5726      2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4815      5727      2      THEN
4816      5728      2          BEGIN
4817      5729      2              RET_LEN[0] = 512;
4818      5730      2              RETURN;
4819      5731      2          END;
4820      5732      2
4821      5733      1      END;

```

```

73 72 65 69 66 69 6C 61 75 51 20 6D 72 6F 46 029B2 P.AGE: .ASCII \Form Qualifiers:\
                                     3A 029C1
                                     029C2
00000010 029C4 P.AGD: .BLKB 2
00000000 029C8 .LONG 16
               .ADDRESS P.AGE

```

BEGIN_FORMAT= P.AGD

				0004 00000	GET_FORM	QUALIFIERS:		
		5E	08	C2	00002	WORD	Save R2	5685
			7E	D4	00005	SUBL2	#8, SP	
			8F	3C	00007	CLRL	CURRENT_LEN	5690
04	AE	0200	8F	3C	00007	MOVZWL	#512, STRING_PTR	5703
	50		08	AC	D0 0000D	MOVL	STR_DESC, R0	5704
08	AE	04	A0	D0	00011	MOVL	4(R0), STRING_PTR+4	
	52		0C	AC	D0 00016	MOVL	RET_LEN, R2	5706
			62	B4	0001A	CLRW	(R2)	
		04	AE	9F	0001C	PUSHAB	STRING_PTR	5711
		04	AE	9F	0001F	PUSHAB	CURRENT_LEN	
		D3	AF	9F	00022	PUSHAB	BEGIN_FORMAT	
00000000G	00		03	FB	00025	CALLS	#3, SYS\$FAO	
	62		6E	A0	0002C	ADDW2	CURRENT_LEN, (R2)	5713
08	AE		6E	C0	0002F	ADDL2	CURRENT_LEN, STRING_PTR+4	5714
04	AE		62	3C	00033	MOVZWL	(R2), STRING_PTR	5715
04	AE	00000200	8F	C3	00037	SUBL3	STRING_PTR, #512, STRING_PTR	
	12		62	B1	00041	CMPL	(R2), #18	5721
			02	1A	00044	BGTRU	1\$	
			62	B4	00046	CLRW	(R2)	5723
0200	8F		62	B1	00048	CMPL	(R2), #512	5726
			05	1B	0004D	BLEQU	2\$	
	62	0200	8F	B0	0004F	MOVW	#512, (R2)	5729
			04	00054	2\$:	RET		5733

; Routine Size: 85 bytes, Routine Base: CODE + 29CC

```
4823 5734 1 %sbttl 'GET_USER_NOTE- Insert a Note into the Page'
4824 5735 1 ++
4825 5736 1 Functional Description:
4826 5737 1 This routine gets a note as specified by the user for the frame.
4827 5738 1
4828 5739 1 Formal Parameters:
4829 5740 1 SCB - Address of the SCB
4830 5741 1 STR_DESC - Desc of String to Return
4831 5742 1 RET_LEN - Return length of Desc.
4832 5743 1
4833 5744 1 Implicit Inputs:
4834 5745 1 none
4835 5746 1
4836 5747 1 Implicit Outputs:
4837 5748 1 none
4838 5749 1
4839 5750 1 Returned Value:
4840 5751 1 none
4841 5752 1
4842 5753 1 Side Effects:
4843 5754 1 none
4844 5755 1 --
4845 5756 1 ROUTINE GET_USER_NOTE (
4846 5757 1 SCB : REF $BLOCK, : SCB
4847 5758 1 STR_DESC : REF VECTOR[2], : Output buffer desc
4848 5759 1 RET_LEN : REF VECTOR [,WORD] : Return length (word)
4849 5760 1 ) : NOVALUE =
4850 5761 2 BEGIN
4851 5762 2 BIND
4852 P 5763 2 NOTE_FULL_FORMAT = $DESCRIPTOR (
4853 5764 2 'NOTE: !AF' ); ! - user note
4854 5765 2 LITERAL
4855 5766 2 K_MIN_NOTE_LEN = 6;
4856 5767 2 LOCAL
4857 5768 2 STATUS;
4858 5769 2
4859 P 5770 2 STATUS = $FAO ( NOTE_FULL_FORMAT,
4860 P 5771 2 RET_LEN[0],
4861 P 5772 2 STR_DESC[0],
4862 P 5773 2 .SCB_SIZE (NOTE),
4863 5774 2 .SCB_ADDR (NOTE)); ! user note
4864 5775 2
4865 5776 2 IF .RET_LEN[0] LEQ K_MIN_NOTE_LEN ! print nothing... no note
4866 5777 2 THEN RET_LEN[0] = 0;
4867 5778 2
4868 5779 2
4869 5780 2 RETURN SS$ _NORMAL;
4870 5781 1 END;
```

```
46 41 21 20 3A 45 54 4F 4E 02A21 P.AGG: .ASCII \NOTE: !AF\
02A2A .BLKB 2
00000009 02A2C P.AGF: .LONG 9
00000000 02A30 .ADDRESS P.AGG
NOTE_FULL_FORMAT= P.AGF
```

```

0000 00000 GET_USER_NOTE:
50      04 AC D0 00002      .WORD      Save nothing      : 5756
      00D8 C0 DD 00006      MOVL      SCB, R0      : 5774
7E      00D4 C0 3C 0000A      PUSHL     216(R0)
      08 AC DD 0000F      MOVZWL    212(R0), -(SP)
      0C AC DD 00012      PUSHL     STR_DESC
      E0 AF 9F 00015      PUSHL     RET_LEN
00000000G 00      05 FB 00018      PUSHAB  NOTE_FULL_FORMAT
      06      0C BC B1 0001F      CALLS    #5, SYSSFAO
      03      1A 00023      CMPW     @RET_LEN, #6
      0C BC B4 00025      BGTRU    1$
      04 00028 1$:      CLRW     @RET_LEN
      RET

```

; Routine Size: 41 bytes, Routine Base: CODE + 2A34


```
4872 5782 1 %sbttl 'GET_RECEIPT_BOX - Insert a "Received Box" into the Page'
4873 5783 1 ++
4874 5784 1 Functional Description:
4875 5785 1 This routine gets a note as specified by the user for the frame.
4876 5786 1
4877 5787 1 Formal Parameters:
4878 5788 1 SCB - Address of the SCB
4879 5789 1 STR_DESC - Desc of String to Return
4880 5790 1 RET_LEN - Return length of Desc.
4881 5791 1
4882 5792 1 Implicit Inputs:
4883 5793 1 none
4884 5794 1
4885 5795 1 Implicit Outputs:
4886 5796 1 none
4887 5797 1
4888 5798 1 Returned Value:
4889 5799 1 none
4890 5800 1
4891 5801 1 Side Effects:
4892 5802 1 none
4893 5803 1 --
4894 5804 1 ROUTINE GET_RECEIPT_BOX (
4895 5805 1 SCB : REF $BBLOCK, : SCB
4896 5806 1 STR_DESC : REF VECTOR[2], : Output buffer desc
4897 5807 1 RET_LEN : REF VECTOR [,WORD] : Return length (word)
4898 5808 1 ) : NOVALUE =
4899 5809 2 BEGIN
4900 5810 2 BIND
4901 5811 2 NOTE132_FORMAT = $DESCRIPTOR (
4902 5812 2
4903 5813 2
4904 5814 2 Received: .....
4905 5815 2
4906 5816 2 Date : .....
4907 5817 2
4908 5818 2 Operator: .....
4909 5819 2
4910 5820 2 ) : VECTOR; ! - receipt box
4911 5821 2
4912 5822 2 CH$MOVE (.NOTE132_FORMAT[0], .NOTE132_FORMAT[1], .STR_DESC[ADDR]);
4913 5823 2 RET_LEN[0] = .NOTE132_FORMAT[0];
4914 5824 2
4915 5825 2 RETURN SS$_NORMAL;
4916 5826 1 END;
```

```
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2B 02A5D P.AGI: .ASCII \+-----+\
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 02A6C
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 02A7B
20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 02A85 .ASCII \! !\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02A94
2E 2E 2E 2E 3A 64 65 76 69 65 63 65 52 20 21 02AA3
2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 02AAD .ASCII \! Received: ..... !\
2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 02ABC
21 20 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 02ACB
```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_RECEIPT_BOX - Insert a "Received Box" into

K 9
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 162
(38)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	21	02AD5
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	02AE4
2E	2E	2E	20	3A	20	20	20	20	20	65	74	61	44	20	21	02AF3
2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	02AFD
					21	20	2E	2E	2E	2E	2E	2E	2E	2E	2E	02B0C
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	21	02B1B
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	02B25
					21	20	20	20	20	20	20	20	20	20	20	02B34
2E	2E	2E	20	3A	72	6F	74	61	72	65	70	4F	20	21	20	02B43
2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	02B4D
					21	20	2E	2E	2E	2E	2E	2E	2E	2E	2E	02B5C
2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	02B6B
2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	02B75
					2B	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	02B84
																02B93
																02B9D

.ASCII \!
:!
.ASCII \! Date :!
:!
.ASCII \!
:!
.ASCII \! Operator:!
:!
.ASCII \+-----+!
:!
:!
:!

00000140 02BA0 P.AGH: .BLKB 3
00000000 02BA4 .LONG 320
.ADDRESS P.AGI

NOTE132_FORMAT= P.AGH

003C 00000 GET_RECEIPT_BOX:

			50	08	AC	D0	00002
04	B0	F1	BF	EF	AF	28	00006
		0C	BC	E8	AF	B0	0000D
						04	00012

.WORD Save R2,R3,R4,R5
MOVL STR_DESC, R0
MOVCL NOTE132_FORMAT, @NOTE132_FORMAT+4, @4(R0)
MOVW NOTE132_FORMAT, @RET_LEN
RET

: 5804
: 5822
: 5823
: 5826

; Routine Size: 19 bytes, Routine Base: CODE + 2BA8

```
4918 5827 1 %sbttl 'GET_RULER_FINE - Insert a fine 'RULER' into the Page'
4919 5828 1 ++
4920 5829 1 Functional Description:
4921 5830 1 This routine gets a fine ruler '1234567890'
4922 5831 1
4923 5832 1 Formal Parameters:
4924 5833 1 SCB - Address of the SCB
4925 5834 1 STR_DESC - Desc of String to Return
4926 5835 1 RET_LEN - Return length of Desc.
4927 5836 1
4928 5837 1 Implicit Inputs:
4929 5838 1 none
4930 5839 1
4931 5840 1 Implicit Outputs:
4932 5841 1 none
4933 5842 1
4934 5843 1 Returned Value:
4935 5844 1 none
4936 5845 1
4937 5846 1 Side Effects:
4938 5847 1 none
4939 5848 1 --
4940 5849 1 ROUTINE GET_RULER_FINE (
4941 5850 1 SCB : REF $BBLOCK, ! SCB
4942 5851 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
4943 5852 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
4944 5853 1 ) : NOVALUE =
4945 5854 2 BEGIN
4946 5855 2 BIND
4947 5856 2 VMS_FORMAT = $DESCRIPTOR (
4948 5857 2 '1234567890'
4949 5858 2 );
4950 5859 2
4951 5860 2 $FAO ( VMS_FORMAT,
4952 5861 2 RET_LEN[0],
4953 5862 2 STR_DESC[0]
4954 5863 2 );
4955 5864 2
4956 5865 2 RETURN SS$_NORMAL;
4957 5866 1 END;
```

```
30 39 38 37 36 35 34 33 32 31 02BBB P.AGK: .ASCII \1234567890\
02BC5 .BLKB 3
0000000A 02BC8 P.AGJ: .LONG 10
00000000 02BCC .ADDRESS P.AGK
VMS_FORMAT= P.AGJ
```

```
0000 0000 GET_RULER_FINE:
08 AC DD 00002 .WORD Save nothing
OC AC DD 00005 PUSHL STR_DESC
ED AF 9F 00008 PUSHAB RET_LEN
03 FB 0000B CALLS VMS_FORMAT
00000000G 00 #3, SYSSFAO
```

5849
5863

SEPARATE
V04-001

Print Symbiont -- separation routines.
GET_RULER_FINE - Insert a fine 'RULER' into the

M 9
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 164
(39)

04 00012

RET

; 5866

; Routine Size: 19 bytes, Routine Base: CODE + 2BD0


```

4959 5867 1 %sbttl 'GET_RULER_COARSE - Insert a coarse 'RULER' into the Page'
4960 5868 1 ++
4961 5869 1 Functional Description:
4962 5870 1 This routine gets a coarse ruler '1...2...3...' for the frame.
4963 5871 1
4964 5872 1 Formal Parameters:
4965 5873 1 SCB - Address of the SCB
4966 5874 1 STR_DESC - Desc of String to Return
4967 5875 1 RET_LEN - Return length of Desc.
4968 5876 1
4969 5877 1 Implicit Inputs:
4970 5878 1 none
4971 5879 1
4972 5880 1 Implicit Outputs:
4973 5881 1 none
4974 5882 1
4975 5883 1 Returned Value:
4976 5884 1 none
4977 5885 1
4978 5886 1 Side Effects:
4979 5887 1 none
4980 5888 1 --
4981 5889 1 ROUTINE GET_RULER_COARSE (
4982 5890 1 SCB : REF $BLOCK, : SCB
4983 5891 1 STR_DESC : REF VECTOR[2], : Output buffer desc
4984 5892 1 RET_LEN : REF VECTOR [,WORD] : Return length (word)
4985 5893 1 ) : NOVALUE =
4986 5894 2 BEGIN
4987 5895 2 BIND
4988 5896 2 VMS_FORMAT = $DESCRIPTOR (
4989 5897 2 :
4990 5898 2 :
4991 5899 2 :
4992 5900 2 :
4993 5901 2 :
4994 5902 2 :
4995 5903 2 :
4996 5904 2 :
4997 5905 2 :
4998 5906 2 :
4999 5907 2 );
5000 5908 2
5001 5909 2 $FAO ( VMS_FORMAT,
5002 5910 2 RET_LEN[0],
5003 5911 2 STR_DESC[0],
5004 5912 2 );
5005 5913 2
5006 5914 2 RETURN SS$_NORMAL;
5007 5915 1 END;

```

31	20	20	20	20	20	20	20	20	20	02BE3	P.AGM:	.ASCII	\	1\
32	20	20	20	20	20	20	20	20	20	02BED		.ASCII	\	2\
33	20	20	20	20	20	20	20	20	20	02BF7		.ASCII	\	3\
34	20	20	20	20	20	20	20	20	20	02C01		.ASCII	\	4\
35	20	20	20	20	20	20	20	20	20	02C0B		.ASCII	\	5\

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_RULER_COARSE - Insert a coarse "RULER" into

B 10

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 166
(40)

36	20	20	20	20	20	20	20	20	20	20	02C15	.ASCII	\	6\	:	
37	20	20	20	20	20	20	20	20	20	20	02C1F	.ASCII	\	7\	:	
38	20	20	20	20	20	20	20	20	20	20	02C29	.ASCII	\	8\	:	
39	20	20	20	20	20	20	20	20	20	20	02C33	.ASCII	\	9\	:	
30	20	20	20	20	20	20	20	20	20	20	02C3D	.ASCII	\	0\	:	
											02C47	.BLKB	1		:	
											00000064	02C48	P.AGL:	.LONG	100	:
											00000000	02C4C		.ADDRESS	P.AGM	:

VMS_FORMAT= P.AGL

0000 00000 GET_RULER_COARSE:

08	AC	DD	00002	.WORD	Save nothing	:	5889
0C	AC	DD	00005	PUSHL	STR_DESC	:	5912
ED	AF	9F	00008	PUSHL	RET_LEN	:	
				PUSHAB	VMS_FORMAT	:	
	03	FB	0000B	CALLS	#3, SYSSFAO	:	
				RET		:	5915
	04		00012			:	

00000000G 00

; Routine Size: 19 bytes. Routine Base: CODE + 2C50

```
5009 5916 1 %sbttl 'GET_FORM_SIZE - Determine the Size of Form Specified'
5010 5917 1 ++
5011 5918 1 Functional Description:
5012 5919 1 This routine determines the which standard form is
5013 5920 1 specified by interrogating the SCB for length and width
5014 5921 1 Standard forms sizes include: 132x66, 132x51, 80x66, 80x51 or
5015 5922 1 40xany_length. Otherwise form size is 'non_std'.
5016 5923 1
5017 5924 1 Formal Parameters:
5018 5925 1 SCB - Address of the SCB
5019 5926 1
5020 5927 1 Implicit Inputs:
5021 5928 1 none
5022 5929 1
5023 5930 1 Implicit Outputs:
5024 5931 1 none
5025 5932 1
5026 5933 1 Returned Value:
5027 5934 1 none
5028 5935 1
5029 5936 1 Side Effects:
5030 5937 1 none
5031 5938 1 --
5032 5939 1 ROUTINE GET_FORM_SIZE (
5033 5940 1 SCB : REF $BBLOCK
5034 5941 1 ): NOVALUE =
5035 5942 2 BEGIN
5036 5943 2
5037 5944 2 SCB[PSM$L_PAGE_WIDTH] = .SCB[PSM$L_FORM_WIDTH];
5038 5945 2 IF .SCB[PSM$L_PAGE_WIDTH] GTRU 200
5039 5946 2 THEN
5040 5947 2 SCB[PSM$L_PAGE_WIDTH] = 200;
5041 5948 2
5042 5949 2 SCB[PSM$L_PAGE_LENGTH] = .SCB[PSM$L_FORM_LENGTH];
5043 5950 2 IF .SCB[PSM$L_PAGE_LENGTH] GTRU 100
5044 5951 2 THEN
5045 5952 2 SCB[PSM$L_PAGE_LENGTH] = 100;
5046 5953 2
5047 5954 2 WHILE .SCB[PSM$L_PAGE_LENGTH] LSSU 40
5048 5955 2 DO
5049 5956 2 SCB[PSM$L_PAGE_LENGTH] =
5050 5957 2 .SCB[PSM$L_PAGE_LENGTH] + .SCB[PSM$L_FORM_LENGTH];
5051 5958 2
5052 5959 1 END;
```

```
0000 0000 GET_FORM_SIZE:
      50      04 AC D0 00002      .WORD      Save nothing
      51      0200 CO 9E 00006      MOVL      SCB, R0
      61      008C CO D0 0000B      MOVAB     512(R0), R1
000000C8 8F 61 D1 00010      MOVL      140(R0), (R1)
      04 1B 00017      CMPL      (R1), #200
      61      C8 8F 9A 00019      BLEQU     1$
      61      C8 8F 9A 00019      MOVZBL    #200, (R1)
```

```
: 5939
: 5944
:
: 5945
:
: 5947
```

; Routine Size: 63 bytes, Routine Base: CODE + 2C63


```
5054 5960 1 %sbttl 'FILL_FRAME - Insert Information into this frame of the Page'
5055 5961 1 ++
5056 5962 1 Functional Description:
5057 5963 1 This procedure inserts a character into an array(frame)
5058 5964 1 until no room left in the frame.
5059 5965 1
5060 5966 1 Formal Parameters:
5061 5967 1 SCB - Address of the SCB
5062 5968 1 CHAR - Descriptor of String to Insert
5063 5969 1 FRAME_PTR - Address of first byte of frame
5064 5970 1 FRAME_LENGTH - Length of frame
5065 5971 1 FRAME_WIDTH - Width of frame
5066 5972 1
5067 5973 1 Implicit Inputs:
5068 5974 1 none
5069 5975 1
5070 5976 1 Implicit Outputs:
5071 5977 1 none
5072 5978 1
5073 5979 1 Returned Value:
5074 5980 1 none
5075 5981 1
5076 5982 1 Side Effects:
5077 5983 1 none
5078 5984 1 --
5079 5985 1 ROUTINE FILL_FRAME (
5080 5986 1 SCB : REF $BBLOCK,
5081 5987 1 CHAR,
5082 5988 1 FRAME_PTR : REF PAGE_ARRAY,
5083 5989 1 FRAME_WIDTH,
5084 5990 1 FRAME_LENGTH
5085 5991 1 ): NOVALUE =
5086 5992 2 BEGIN
5087 5993 2
5088 5994 2 LOCAL PTR : REF PAGE_ARRAY,
5089 5995 2 LOC_FRAME_LENGTH,
5090 5996 2 LOC_FRAME_WIDTH ;
5091 5997 2
5092 5998 2 ! Check for dumb calls
5093 5999 2
5094 6000 3 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)
5095 6001 2 THEN
5096 6002 2 RETURN;
5097 6003 2 IF (.FRAME_LENGTH GTR .SCB[PSM$PAGE_LENGTH]) OR
5098 6004 3 (.FRAME_WIDTH GTR .SCB[PSM$PAGE_WIDTH])
5099 6005 2 THEN
5100 6006 2 RETURN;
5101 6007 2
5102 6008 2 ! Check page boundary conditions
5103 6009 2
5104 6010 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
5105 6011 2 IF .LOC_FRAME_LENGTH GTR .SCB[PSM$PAGE_LENGTH]
5106 6012 2 THEN
5107 6013 2 LOC_FRAME_LENGTH = .SCB[PSM$PAGE_LENGTH]; ! stay in page bounds
5108 6014 2
5109 6015 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
5110 6016 2 IF .LOC_FRAME_WIDTH GTR .SCB[PSM$PAGE_WIDTH]
```

```
5111 6017 2 THEN
5112 6018 2 LOC_FRAME_WIDTH = .SCB[PSM$L_PAGE_WIDTH];
5113 6019 2
5114 6020 2
5115 6021 2 PTR = FRAME_PTR[0,0,.SCB[PSM$L_PAGE_WIDTH]];
5116 6022 2
5117 6023 2
5118 6024 2 DECR L FROM .LOC_FRAME_LENGTH TO 1 DO
5119 6025 2 BEGIN
5120 6026 2 CH$FILL( .CHAR, .LOC_FRAME_WIDTH,.PTR);
5121 6027 2
5122 6028 2 PTR = .PTR + .SCB[PSM$L_PAGE_WIDTH];
5123 6029 2
5124 6030 2
5125 6031 1 END;
```

! stay in page bounds

! Address calc. is based
! on Form Width

				03FC 00000 FILL_FRAME:			
				WORD	Save R2,R3,R4,R5,R6,R7,R8,R9	5985	
	52	14	AC D0 00002	MOVL	FRAME_LENGTH, R2	6000	
			4C 15 00006	BLEQ	5\$		
		10	AC D5 00008	TSTL	FRAME_WIDTH		
			47 15 0000B	BLEQ	5\$		
	50	04	AC D0 0000D	MOVL	SCB, R0	6003	
	51	01F8	C0 D0 00011	MOVL	504(R0), R1		
	51		52 D1 00016	CMPL	R2, R1		
			39 14 00019	BGTR	5\$		
	0200	C0	10 AC D1 0001B	CMPL	FRAME_WIDTH, 512(R0)	6004	
			31 14 00021	BGTR	5\$		
	56		52 D0 00023	MOVL	R2, LOC_FRAME_LENGTH	6010	
	51		56 D1 00026	CMPL	LOC_FRAME_LENGTH, R1	6011	
			03 15 00029	BLEQ	1\$		
	56		51 D0 0002B	MOVL	R1, LOC_FRAME_LENGTH	6013	
	59	10	AC D0 0002E 1\$:	MOVL	FRAME_WIDTH, LOC_FRAME_WIDTH	6015	
	58	0200	C0 D0 00032	MOVL	512(R0), R8	6016	
	58		59 D1 00037	CMPL	LOC_FRAME_WIDTH, R8		
			03 15 0003A	BLEQ	2\$		
	59		58 D0 0003C	MOVL	R8, LOC_FRAME_WIDTH	6018	
	57	0C	AC D0 0003F 2\$:	MOVL	FRAME_PTR, PTR	6021	
			56 D6 00043	INCL	L	6026	
			0A 11 00045	BRB	4\$		
59	08	AC	6E 00 2C 00047 3\$:	MOVC5	#0, (SP), CHAR, LOC_FRAME_WIDTH, (PTR)		
			67 0004D				
	57		58 C0 0004E	ADDL2	R8, PTR	6028	
	F3		56 F5 00051 4\$:	SOBGTR	L, 3\$	6024	
			04 00054 5\$:	RET		6031	

; Routine Size: 85 bytes, Routine Base: CODE + 2CA2

```
5127 6032 1 %sbttl 'SCROLL_FRAME - Insert Information into this Frame of the Page'
5128 6033 1 **
5129 6034 1 Functional Description:
5130 6035 1 This procedure inserts a string into an array(frame) repeatedly
5131 6036 1 until no room is left in the frame.
5132 6037 1
5133 6038 1 Formal Parameters:
5134 6039 1 SCB - Address of the SCB
5135 6040 1 CHAR STRING - Descriptor of String to Insert
5136 6041 1 FRAME_PTR - Address of first byte of frame
5137 6042 1 FRAME_LENGTH - Length of frame
5138 6043 1 FRAME_WIDTH - Width of frame
5139 6044 1
5140 6045 1 Implicit Inputs:
5141 6046 1 none
5142 6047 1
5143 6048 1 Implicit Outputs:
5144 6049 1 none
5145 6050 1
5146 6051 1 Returned Value:
5147 6052 1 none
5148 6053 1
5149 6054 1 Side Effects:
5150 6055 1 none
5151 6056 1 --
5152 6057 1 ROUTINE SCROLL_FRAME (
5153 6058 1 SCB : REF $BLOCK,
5154 6059 1 CHAR STRING : REF VECTOR[2],
5155 6060 1 FRAME_PTR : REF PAGE_ARRAY,
5156 6061 1 FRAME_WIDTH ,
5157 6062 1 FRAME_LENGTH ,
5158 6063 1 ) : NOVALUE =
5159 6064 2 BEGIN
5160 6065 2
5161 6066 2 LOCAL PTR : REF PAGE_ARRAY,
5162 6067 2 LOC_FRAME_LENGTH,
5163 6068 2 LOC_FRAME_WIDTH ,
5164 6069 2 TEMP_PTR ,
5165 6070 2 START_CNT : INITIAL (0),
5166 6071 2 CHARS ,
5167 6072 2 NUM_CHARS : INITIAL (0);
5168 6073 2
5169 6074 2 ! Check for dumb calls
5170 6075 2 !
5171 6076 2 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0) OR
5172 6077 2 (.FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]) OR
5173 6078 2 (.FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH])
5174 6079 2 THEN
5175 6080 2 RETURN;
5176 6081 2
5177 6082 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
5178 6083 2
5179 6084 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
5180 6085 2
5181 6086 2 PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
5182 6087 2 TEMP_PTR = .CHAR_STRING[ADDR];
5183 6088 2 CHARS = CH$PTR(.TEMP_PTR);
```

```
5184 6089 2
5185 6090 2 INCR L FROM 0 TO (.LOC_FRAME_LENGTH-1) DO
5186 6091 3 BEGIN
5187 6092 3 PTR = FRAME_PTR[0,0,SCB[PSMSL_PAGE_WIDTH]] ! Address calc. is based
5188 6093 3 + (.L*SCB[PSMSL_PAGE_WIDTH]); ! on Form Width
5189 6094 3
5190 6095 3 ! Move the rest of the string into the beginning of the next frame
5191 6096 3
5192 6097 4 IF (.NUM_CHARS LEQ .CHAR_STRING[SIZE])
5193 6098 4 AND (.CHAR_STRING[SIZE]-.NUM_CHARS) LSS .LOC_FRAME_WIDTH
5194 6099 3 THEN ! Scroll it
5195 6100 4 BEGIN
5196 6101 4 TEMP_PTR = .CHAR_STRING[ADDR] + .NUM_CHARS; ! move remainder of str.
5197 6102 4 CH$MOVE(.CHAR_STRING[SIZE]-.NUM_CHARS, .TEMP_PTR, .PTR);
5198 6103 4 PTR = .PTR + (.CHAR_STRING[SIZE]-.NUM_CHARS);
5199 6104 4 ! incr by no. inserted
5200 6105 4 START_CNT = .CHAR_STRING[SIZE] - .NUM_CHARS;
5201 6106 4 TEMP_PTR = .CHAR_STRING[ADDR];
5202 6107 3 END;
5203 6108 3
5204 6109 3 INCR I FROM .START_CNT TO .LOC_FRAME_WIDTH BY .CHAR_STRING[SIZE] DO
5205 6110 4 BEGIN
5206 6111 5 IF .CHAR_STRING[SIZE] GEQ (.LOC_FRAME_WIDTH - .I)
5207 6112 4 THEN
5208 6113 5 NUM_CHARS = (.LOC_FRAME_WIDTH - .I)
5209 6114 4 ELSE
5210 6115 4 NUM_CHARS = .CHAR_STRING[SIZE];
5211 6116 4
5212 6117 4 CH$MOVE(.NUM_CHARS, .TEMP_PTR, .PTR);
5213 6118 4
5214 6119 4 PTR = .PTR + .NUM_CHARS;
5215 6120 3 END;
5216 6121 2
5217 6122 1 END;
```

OFFC 00000 SCROLL_FRAME:						
				.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	6057
5E		0C	C2 00002	SUBL2	#12, SP	6064
		59	D4 00005	CLRL	START CNT	
		7E	D4 00007	CLRL	NUM_CHARS	
51	14	AC	D0 00009	MOVL	FRAME_LENGTH, R1	6076
		03	15 0000D	BLEQ	1\$	
	10	AC	D5 0000F	TSTL	FRAME_WIDTH	
		01	14 00012 1\$:	BGTR	2\$	
			04 00014	RET		
50	04	AC	D0 00015 2\$:	MOVL	SCB, R0	6077
01F8	C0	51	D1 00019	CMPL	R1, 504(R0)	
		0A	14 0001E	BGTR	3\$	
50	04	AC	D0 00020	MOVL	SCB, R0	6078
0200	C0	AC	D1 00024	CMPL	FRAME_WIDTH, 512(R0)	
		01	15 0002A 3\$:	BLEQ	4\$	
			04 0002C	RET		
0C	AE	51	D0 0002D 4\$:	MOVL	R1, LOC_FRAME_LENGTH	6082

SEPARATE
V04-001

Print Symbiont -- separation routines
SCROLL_FRAME - Insert Information into this fra

1 10
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 173
(43)

SE
V0

		58	10	AC	D0	00031	MOVL	FRAME_WIDTH, LOC_FRAME_WIDTH	6084	
	04	AE	0C	AC	D0	00035	MOVL	FRAME_PTR, PTR	6086	
		5A	08	AC	D0	0003A	MOVL	CHAR_STRING, R10	6087	
		5B	04	AA	D0	0003E	MOVL	4(R10), TEMP_PTR		
		50		5B	D0	00042	MOVL	TEMP_PTR, CHARS	6088	
	08	AE	04	AC	D0	00045	MOVL	SCB, -8(SP)	6093	
		57		01	CE	0004A	MNEGL	#1, L	6109	
				61	11	0004D	BRB	11\$		
51	08	AE	00000200	8F	C1	0004F	5\$: ADDL3	#512, 8(SP), R1	6093	
50		57		61	C5	00058	MULL3	(R1), L, R0		
	04	AE		0C	BC40	9E	0005C	MOVAB	@FRAME_PTR[R0], PTR	
		6A		6E	D1	00062	CMPL	NUM_CHARS, (R10)	6097	
				22	14	00065	BGTR	6\$		
56		6A		6E	C3	00067	SUBL3	NUM_CHARS, (R10), R6	6098	
		58		56	D1	0006B	CMPL	R6, -LOC_FRAME_WIDTH		
				19	18	0006E	BGEQ	6\$		
5B	04	AA		6E	C1	00070	ADDL3	NUM_CHARS, 4(R10), TEMP_PTR	6101	
56		6A		6E	C3	00075	SUBL3	NUM_CHARS, (R10), R6	6102	
04	BE	6B		56	28	00079	MOVC3	R6, -(TEMP_PTR), @PTR		
	04	AE		56	C0	0007E	ADDL2	R6, PTR	6103	
		59		56	D0	00082	MOVL	R6, START_CNT	6105	
		5B	04	AA	D0	00085	MOVL	4(R10), TEMP_PTR	6106	
		56		59	D0	00089	6\$: MOVL	START_CNT, I	6117	
				1D	11	0008C	BRB	10\$		
50		58		56	C3	0008E	7\$: SUBL3	I, LOC_FRAME_WIDTH, R0	6111	
		50		6A	D1	00092	CMPL	(R10), R0		
				05	19	00095	BLSS	8\$		
		6E		50	D0	00097	MOVL	R0, NUM_CHARS	6113	
				03	11	0009A	BRB	9\$		
		6E		6A	D0	0009C	8\$: MOVL	(R10), NUM_CHARS	6115	
04	BE	6B		6E	28	0009F	9\$: MOVC3	NUM_CHARS, -(TEMP_PTR), @PTR	6117	
	04	AE		6E	C0	000A4	ADDL2	NUM_CHARS, PTR	6119	
		56		6A	C0	000A8	ADDL2	(R10), I	6109	
		58		56	D1	000AB	10\$: CMPL	I, LOC_FRAME_WIDTH		
				DE	15	000AE	BLEQ	7\$		
9A		57	0C	AE	F2	000B0	11\$: AOBLSS	LOC_FRAME_LENGTH, L, 5\$	6090	
				04	000B5		RET		6122	

; Routine Size: 182 bytes, Routine Base: CODE + 2CF7

```
5219 6123 1 %sbttl 'MOVE_FRAME - Move Information into this Frame of the Page'
5220 6124 1 ++
5221 6125 1 Functional Description:
5222 6126 1 This procedure inserts a string(frame) into an array(frame).
5223 6127 1 Insertion continues until either no more string or no more room.
5224 6128 1
5225 6129 1 Formal Parameters:
5226 6130 1 SCB - Address of the SCB
5227 6131 1 CHAR_STRING - Descriptor of String to Insert
5228 6132 1 FRAME_PTR - Address of first byte of Frame
5229 6133 1 FRAME_LENGTH - Length of Frame
5230 6134 1 FRAME_WIDTH - Width of Frame
5231 6135 1
5232 6136 1 Implicit Inputs:
5233 6137 1 none
5234 6138 1
5235 6139 1 Implicit Outputs:
5236 6140 1 none
5237 6141 1
5238 6142 1 Returned Value:
5239 6143 1 none
5240 6144 1
5241 6145 1 Side Effects:
5242 6146 1 Truncation is possible.
5243 6147 1
5244 6148 1 --
5245 6149 1 ROUTINE MOVE_FRAME (
5246 6150 1 SCB : REF $BLOCK,
5247 6151 1 CHAR_STRING : REF VECTOR[2],
5248 6152 1 FRAME_PTR : REF PAGE_ARRAY,
5249 6153 1 FRAME_WIDTH , ! Number of Columns
5250 6154 1 FRAME_LENGTH ! Number of Rows
5251 6155 1 ) : NOVALUE =
5252 6156 2 BEGIN
5253 6157 2
5254 6158 2 LOCAL PTR : REF PAGE_ARRAY,
5255 6159 2 LOC_FRAME_LENGTH,
5256 6160 2 LOC_FRAME_WIDTH,
5257 6161 2 STR_PTR,
5258 6162 2 CURR_SIZE,
5259 6163 2 NUM_CHARS;
5260 6164 2
5261 6165 2 ! Check for dumb calls
5262 6166 2
5263 6167 2 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)
5264 6168 2 THEN
5265 6169 2 RETURN;
5266 6170 2 IF (.FRAME_LENGTH GTR .SCB[PSM$L_PAGE_LENGTH]) OR
5267 6171 2 (.FRAME_WIDTH GTR .SCB[PSM$L_PAGE_WIDTH])
5268 6172 2 THEN
5269 6173 2 RETURN;
5270 6174 2
5271 6175 2
5272 6176 2 ! Check page boundary conditions
5273 6177 2
5274 6178 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
5275 6179 2 IF .LOC_FRAME_LENGTH GTR .SCB[PSM$L_PAGE_LENGTH]
```

```
5276 6180 2 THEN
5277 6181 2 LOC_FRAME_LENGTH = .SCB[PSMSL_PAGE_LENGTH]; ! stay in page bounds
5278 6182 2
5279 6183 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
5280 6184 2 IF .LOC_FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH]
5281 6185 2 THEN
5282 6186 2 LOC_FRAME_WIDTH = .SCB[PSMSL_PAGE_WIDTH]; ! stay in page bounds
5283 6187 2
5284 6188 2 ! Get string info
5285 6189 2 CURR_SIZE = .CHAR_STRING[SIZE];
5286 6190 2 PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
5287 6191 2 STR_PTR = .CHAR_STRING[ADDR];
5288 6192 2
5289 6193 2 ! Do a quick fill of the frame
5290 6194 2 FILL_FRAME (.SCB,
5291 6195 2 XCHAR(32), ! fill with blanks
5292 6196 2 FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
5293 6197 2 .LOC_FRAME_WIDTH,
5294 6198 2 .LOC_FRAME_LENGTH);
5295 6199 2
5296 6200 2 DECRU L FROM .LOC_FRAME_LENGTH TO 1 DO
5297 6201 2 BEGIN
5298 6202 2 IF .CURR_SIZE GEQ .LOC_FRAME_WIDTH THEN
5299 6203 2 NUM_CHARS = .LOC_FRAME_WIDTH
5300 6204 2 ELSE
5301 6205 2 NUM_CHARS = .CURR_SIZE;
5302 6206 2
5303 6207 2 CH$MOVE(.NUM_CHARS, .STR_PTR, .PTR);
5304 6208 2
5305 6209 2 PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH]; ! Address calc. is based
5306 6210 2 ! on Form Width
5307 6211 2 STR_PTR = .STR_PTR + .LOC_FRAME_WIDTH;
5308 6212 2
5309 6213 2 IF .LOC_FRAME_WIDTH GTRU .CURR_SIZE THEN
5310 6214 2 EXITLOOP;
5311 6215 2
5312 6216 2 CURR_SIZE = .CURR_SIZE - .NUM_CHARS; ! Decrease string size
5313 6217 2 END;
5314 6218 2 1 END;
```

OFFC 00000 MOVE_FRAME:							
5E		04	C2	00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	6149
51	14	AC	D0	00005	SUBL2	#4, SP	
		7C	15	00009	MOVL	FRAME_LENGTH, R1	6167
	10	AC	D5	0000B	BLEQ	7\$	
		77	15	0000E	TSTL	FRAME_WIDTH	
57	04	AC	D0	00010	BLEQ	7\$	
50	01F8	C7	D0	00014	MOVL	SCB, R7	6170
50		51	D1	00019	MOVL	504(R7), R0	
		69	14	0001C	CMPL	R1, R0	
0200	C7	10	AC	D1	BGTR	7\$	
		61	14	00024	CMPL	FRAME_WIDTH, 512(R7)	6171
					BGTR	7\$	

SEPARATE
V04-001

Print Symbiont -- separation routines

MOVE_FRAME - Move Information into this Frame

L 10

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742

[PRTSMB.SRC]SEPARATE.B32;2

Page 176

(44)

	52		51	D0	00026	MOVL	R1, LOC_FRAME_LENGTH	:	6178	
	50		52	D1	00029	CMPL	LOC_FRAME_LENGTH, R0	:	6179	
			03	15	0002C	BLEQ	1\$:		
	52		50	D0	0002E	MOVL	R0, LOC_FRAME_LENGTH	:	6181	
	58	10	AC	D0	00031	1\$:	MOVL	FRAME_WIDTH, LOC_FRAME_WIDTH	:	6183
0200	C7		58	D1	00035	CMPL	LOC_FRAME_WIDTH, 512(R7)	:	6184	
			05	15	0003A	BLEQ	2\$:		
	58	0200	C7	D0	0003C	MOVL	512(R7), LOC_FRAME_WIDTH	:	6186	
	50	08	AC	D0	00041	2\$:	MOVL	CHAR_STRING, R0	:	6189
	6E	0C	AC	D0	00045	MOVL	FRAME_PTR, PTR	:	6190	
	5A		60	7D	00049	MOVQ	(R0), CURR_SIZE	:	6189	
			52	DD	0004C	PUSHL	LOC_FRAME_LENGTH	:	6198	
			58	DD	0004E	PUSHL	LOC_FRAME_WIDTH	:	6197	
		0C	AC	DD	00050	PUSHL	FRAME_PTR	:	6196	
			20	DD	00053	PUSHL	#32	:		
			57	DD	00055	PUSHL	R7	:		
FE99	CF		05	FB	00057	CALLS	#5, FILL_FRAME	:		
	56		52	D0	0005C	MOVL	LOC_FRAME_LENGTH, L	:	6200	
			24	11	0005F	BRB	6\$:		
	58		5A	D1	00061	3\$:	CMPL	CURR_SIZE, LOC_FRAME_WIDTH	:	6202
			05	19	00064	BLSS	4\$:		
	59		58	D0	00066	MOVL	LOC_FRAME_WIDTH, NUM_CHARS	:	6203	
			03	11	00069	BRB	5\$:		
	59		5A	D0	0006B	4\$:	MOVL	CURR_SIZE, NUM_CHARS	:	6205
00	BE		59	28	0006E	5\$:	MOVCL	NUM_CHARS, (STR_PTR), @PTR	:	6207
	6B	0200	C7	C0	00073	ADDL2	512(R7), PTR	:	6209	
	6E		58	C0	00078	ADDL2	LOC_FRAME_WIDTH, STR_PTR	:	6211	
	5B		58	D1	0007B	CMPL	LOC_FRAME_WIDTH, CURR_SIZE	:	6213	
	5A		07	1A	0007E	BGTRU	7\$:		
			59	C2	00080	SUBL2	NUM_CHARS, CURR_SIZE	:	6216	
			56	D7	00083	DECL	L	:	6200	
			DA	12	00085	6\$:	BNEQ	3\$:	
			04	00087	7\$:	RET		:	6218	

; Routine Size: 136 bytes, Routine Base: CODE + 2DAD


```
5316 6219 1 %sbttl 'INSERT_FRAME - Insert Information into this Frame of the Page'
5317 6220 1 **
5318 6221 1 Functional Description:
5319 6222 1 This procedure inserts a string into an array(frame).
5320 6223 1 Insertion continues until either no more string or no more room
5321 6224 1 Delimiting characters are used to correctly parse the string
5322 6225 1 prior to insertion.
5323 6226 1
5324 6227 1 Formal Parameters:
5325 6228 1 SCB - Address of the SCB
5326 6229 1 CHAR_STRING - Descriptor of String to Insert
5327 6230 1 FRAME_PTR - Address of first byte of Frame
5328 6231 1 FRAME_LENGTH - Length of Frame
5329 6232 1 FRAME_WIDTH - Width of Frame
5330 6233 1
5331 6234 1 Implicit Inputs:
5332 6235 1 none
5333 6236 1
5334 6237 1 Implicit Outputs:
5335 6238 1 none
5336 6239 1
5337 6240 1 Returned Value:
5338 6241 1 none
5339 6242 1
5340 6243 1 Side Effects:
5341 6244 1 Truncation is possible.
5342 6245 1
5343 6246 1 --
5344 6247 1 ROUTINE INSERT_FRAME (
5345 6248 1 SCB : REF $BLOCK,
5346 6249 1 CHAR_STRING : REF VECTOR[2],
5347 6250 1 FRAME_PTR : REF PAGE_ARRAY,
5348 6251 1 FRAME_WIDTH , Number of Columns
5349 6252 1 FRAME_LENGTH ! Number of Rows
5350 6253 1 ) : NOVALUE =
5351 6254 2 BEGIN
5352 6255 2 BUILTIN AP; ! just in case truncation occurred ... don't delimit
5353 6256 2
5354 6257 2 LITERAL
5355 6258 2 LEADING = 0;
5356 6259 2 TRAILING = 1;
5357 6260 2
5358 6261 2 LOCAL PTR : REF PAGE_ARRAY,
5359 6262 2 LOC_FRAME_LENGTH,
5360 6263 2 LOC_FRAME_WIDTH,
5361 6264 2 STR_PTR,
5362 6265 2 CURR_SIZE,
5363 6266 2 DUM_LEN,
5364 6267 2 NUM_CHARS;
5365 6268 2
5366 6269 2 ! Check for dumb calls
5367 6270 2
5368 6271 3 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)
5369 6272 3 THEN
5370 6273 3 RETURN;
5371 6274 2 IF (.FRAME_LENGTH GTR .SCB[PSM$PAGE_LENGTH]) OR
5372 6275 3 (.FRAME_WIDTH GTR .SCB[PSM$PAGE_WIDTH])
```

```
5373 6276 2 THEN
5374 6277 2 RETURN;
5375 6278 2
5376 6279 2 ! Check page boundary conditions
5377 6280 2
5378 6281 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
5379 6282 2 IF .LOC_FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]
5380 6283 2 THEN
5381 6284 2 LOC_FRAME_LENGTH = .SCB[PSMSL_PAGE_LENGTH]; ! stay in page bounds
5382 6285 2
5383 6286 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
5384 6287 2 IF .LOC_FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH]
5385 6288 2 THEN
5386 6289 2 LOC_FRAME_WIDTH = .SCB[PSMSL_PAGE_WIDTH]; ! stay in page bounds
5387 6290 2
5388 6291 2 ! Get string info
5389 6292 2 CURR_SIZE = .CHAR_STRING[SIZE];
5390 6293 2 PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
5391 6294 2 STR_PTR = .CHAR_STRING[ADDR];
5392 6295 2
5393 6296 2 ! Do a quick fill of the frame
5394 6297 2 FILL_FRAME (.SCB,
5395 6298 2 %CHAR(32), ! fill with blanks
5396 6299 2 FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
5397 6300 2 .LOC_FRAME_WIDTH,
5398 6301 2 .LOC_FRAME_LENGTH);
5399 6302 2
5400 6303 2 DECR L FROM .LOC_FRAME_LENGTH TO 1 DO
5401 6304 2 BEGIN
5402 6305 2 IF .CURR_SIZE GEQ .LOC_FRAME_WIDTH THEN
5403 6306 2 NUM_CHARS = DELIMIT_STRING(.STR_PTR,%CHAR(32),.LOC_FRAME_WIDTH)
5404 6307 2 ELSE
5405 6308 2 NUM_CHARS = .CURR_SIZE;
5406 6309 2
5407 6310 2 DISCARD (LEADING,%C' ',.STR_PTR,.NUM_CHARS,.NUM_CHARS,.STR_PTR);
5408 6311 2 CHSMOVE(.NUM_CHARS,.STR_PTR,.PTR);
5409 6312 2
5410 6313 2 PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH]; ! Address calc. is based
5411 6314 2 STR_PTR = .STR_PTR + .NUM_CHARS; ! on Form Width
5412 6315 2
5413 6316 2 IF .LOC_FRAME_WIDTH GEQ .CURR_SIZE THEN
5414 6317 2 EXITLOOP;
5415 6318 2
5416 6319 2 CURR_SIZE = .CURR_SIZE - .NUM_CHARS; ! Decrease string size
5417 6320 2 END;
5418 6321 2
5419 6322 2
5420 6323 2 IF .STR_PTR LSS (.CHAR_STRING[ADDR] + .CHAR_STRING[SIZE])
5421 6324 2 THEN ! truncation occurred
5422 6325 2 CALLG (.AP, MOVE_FRAME); ! dont delimit..just move str.
5423 6326 2 END;
```

OFFC 00000 INSERT_FRAME:

	5E		08	C2	00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	6247
	51	14	AC	D0	00005	SUBL2	#8, SP	
			03	15	00009	MOVL	FRAME_LENGTH, R1	6271
		10	AC	D5	0000B	BLEQ	1\$	
			01	14	0000E	TSTL	FRAME_WIDTH	
				04	00010	BGTR	2\$	
						RET		
	58	04	AC	D0	00011	MOVL	SCB, R8	6274
	50	01F8	C8	D0	00015	MOVL	504(R8), R0	
	50		51	D1	0001A	CMPL	R1, R0	
			06	14	0001D	BGTR	3\$	
0200	C8	10	AC	D1	0001F	CMPL	FRAME_WIDTH, 512(R8)	6275
			01	15	00025	BLEQ	4\$	
				04	00027	RET		
	52		51	D0	00028	MOVL	R1, LOC FRAME LENGTH	6281
	50		52	D1	0002B	CMPL	LOC_FRAME_LENGTH, R0	6282
			03	15	0002E	BLEQ	5\$	
	52		50	D0	00030	MOVL	R0, LOC FRAME LENGTH	6284
	59	10	AC	D0	00033	MOVL	FRAME_WIDTH, LOC_FRAME_WIDTH	6286
0200	C8		59	D1	00037	CMPL	LOC_FRAME_WIDTH, 512(R8)	6287
			05	15	0003C	BLEQ	6\$	
	59	0200	C8	D0	0003E	MOVL	512(R8), LOC_FRAME_WIDTH	6289
	57	08	AC	D0	00043	MOVL	CHAR_STRING, R7	6292
	5A		67	D0	00047	MOVL	(R7), CURR_SIZE	
	5B	0C	AC	D0	0004A	MOVL	FRAME_PTR, PTR	6293
	6E	04	A7	D0	0004E	MOVL	4(R7), STR_PTR	6294
			52	DD	00052	PUSHL	LOC_FRAME_LENGTH	6301
			59	DD	00054	PUSHL	LOC_FRAME_WIDTH	6300
		0C	AC	DD	00056	PUSHL	FRAME_PTR	6299
			20	DD	00059	PUSHL	#32	
			58	DD	0005B	PUSHL	R8	
FE0B	CF		05	FB	0005D	CALLS	#5, FILL_FRAME	
	56	01	A2	9E	00062	MOVAB	1(R2), L	6303
			47	11	00066	BRB	10\$	
	59		5A	D1	00068	CMPL	CURR_SIZE, LOC_FRAME_WIDTH	6305
			12	19	0006B	BLSS	8\$	
			59	DD	0006D	PUSHL	LOC_FRAME_WIDTH	6306
			20	DD	0006F	PUSHL	#32	
		08	AE	DD	00071	PUSHL	STR_PTR	
0000V	CF		03	FB	00074	CALLS	#3, DELIMIT STRING	
04	AE		50	D0	00079	MOVL	R0, NUM_CHARS	
			04	11	0007D	BRB	9\$	
04	AE		5A	D0	0007F	MOVL	CURR_SIZE, NUM_CHARS	6308
			5E	DD	00083	PUSHL	SP	6310
		08	AE	9F	00085	PUSHAB	NUM_CHARS	
		0C	AE	DD	00088	PUSHL	NUM_CHARS	
		0C	AE	DD	0008B	PUSHL	STR_PTR	
			20	DD	0008E	PUSHL	#32	
			7E	D4	00090	CLRL	-(SP)	
	0000V	CF	06	FB	00092	CALLS	#6, DISCARD	
6B	00	04	AE	28	00097	MOVCL	NUM_CHARS, @STR_PTR, (PTR)	6311
	BE	0200	C8	C0	0009D	ADDL2	512(R8), PTR	6313
	5B	04	AE	C0	000A2	ADDL2	NUM_CHARS, STR_PTR	6315
	6E		59	D1	000A6	CMPL	LOC_FRAME_WIDTH, CURR_SIZE	6317
	5A		07	18	000A9	BGEQ	11\$	
		04	AE	C2	000AB	SUBL2	NUM_CHARS, CURR_SIZE	6320
	B6		56	F5	000AF	SOBGTR	L, 7\$	6303

```
Print Symbiont -- separation routines
INSERT_FRAME - Insert Information into this fra
```

C 11
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 180
(45)

57 04 A7
 57

FEB7 CF

67	C1	000B2	11\$:
6E	D1	000B7	
05	18	000BA	
6C	FA	000BC	
	04	000C1	12\$:

```
ADDL3      (R7), 4(R7), R7
CML        STR_PTR, R7
BGEQ       12$
CALLG      (AP), MOVE_FRAME
RET
```

6323
6325
6326

; Routine Size: 194 bytes, Routine Base: CODE + 2E35

SE
VO


```
5425 6327 1 %sbttl 'CENTER_FRAME - Insert String Information into the Center of this Frame'
5426 6328 1 **
5427 6329 1 Functional Description:
5428 6330 1 This procedure inserts a string into the center of an array(frame).
5429 6331 1 Insertion continues until either no more string or no more room
5430 6332 1 The idea is to center the string within the frame.
5431 6333 1 1) If the string is shorter than the entire frame then
5432 6334 1 center the string in the frame.
5433 6335 1 2) If the string is longer than the entire frame then
5434 6336 1 insert as much of the string as possible.
5435 6337 1 3) Use an assumed pad of blanks for beginning and end of string.
5436 6338 1
5437 6339 1
5438 6340 1 Formal Parameters:
5439 6341 1 SCB - Address of the SCB
5440 6342 1 CHAR_STRING - Descriptor of String to Insert
5441 6343 1 FRAME_PTR - Address of first byte of frame
5442 6344 1 FRAME_LENGTH - Length of frame
5443 6345 1 FRAME_WIDTH - Width of frame
5444 6346 1
5445 6347 1 Implicit Inputs:
5446 6348 1 none
5447 6349 1
5448 6350 1 Implicit Outputs:
5449 6351 1 none
5450 6352 1
5451 6353 1 Returned Value:
5452 6354 1 none
5453 6355 1
5454 6356 1 Side Effects:
5455 6357 1 Truncation is possible.
5456 6358 1
5457 6359 1 --
5458 6360 1 ROUTINE CENTER_FRAME (
5459 6361 1 SCB : REF $BBLOCK,
5460 6362 1 CHAR_STRING : REF VECTOR[2],
5461 6363 1 FRAME_PTR : REF PAGE_ARRAY,
5462 6364 1 FRAME_WIDTH : Number of Columns
5463 6365 1 FRAME_LENGTH : Number of Rows
5464 6366 1 ) : NOVALUE =
5465 6367 2 BEGIN
5466 6368 2 LITERAL K_MAX_BUFFER_SIZE = 512, ! maximum possible buffer size
5467 6369 2 K_PAD_LEN = 2; ! length to pad the string
5468 6370 2
5469 6371 2 LOCAL PTR : REF PAGE_ARRAY, ! points to col 0, variable row
5470 6372 2 LOC_FRAME_LENGTH, ! local count of frame_length
5471 6373 2 LOC_FRAME_WIDTH, ! local count of frame_width
5472 6374 2 PAD_CHAR, ! pad character is assumed a space
5473 6375 2 PTR_OFFSET : SIGNED, ! pos/neg offset to origin
5474 6376 2 STR_PTR, ! points to str position in frame
5475 6377 2 STR_SIZE, ! num char left in string
5476 6378 2 BUFFER : VECTOR [512,byte], ! variable buff for pad and string
5477 6379 2 STR_DESC : VECTOR [2]; ! desc of string
5478 6380 2
5479 6381 2 ! Check for dumb calls. Frame dimensions must be writable
5480 6382 2
5481 6383 3 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)
```

```
5482 6384 2 THEN
5483 6385 RETURN;
5484 6386 IF (.FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]) OR
5485 6387 (.FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH])
5486 6388 THEN
5487 6389 RETURN;
5488 6390 ! String must not be zero !!
5489 6391 !
5490 6392 IF .CHAR_STRING[SIZE] EQL 0 THEN RETURN SS$_NORMAL;
5491 6393 !
5492 6394 ! Check page boundary conditions
5493 6395 !
5494 6396 LOC_FRAME_LENGTH = .FRAME_LENGTH;
5495 6397 IF .LOC_FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]
5496 6398 THEN
5497 6399 LOC_FRAME_LENGTH = .SCB[PSMSL_PAGE_LENGTH]; ! stay in page bounds
5498 6400 !
5499 6401 LOC_FRAME_WIDTH = .FRAME_WIDTH;
5500 6402 IF .LOC_FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH]
5501 6403 THEN
5502 6404 LOC_FRAME_WIDTH = .SCB[PSMSL_PAGE_WIDTH]; ! stay in page bounds
5503 6405 !
5504 6406 ! Get string into padding buffer if enough room
5505 6407 !
5506 6408 STR_DESC[SIZE] = %ALLOCATION (BUFFER); ! length of string and pad chars
5507 6409 STR_DESC[ADDR] = BUFFER; ! ptr into pad&string buffer
5508 6410 PAD_CHAR = %CHAR(32,32);
5509 6411 PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]]; ! init
5510 6412 !
5511 6413 ! Pad the string if there is enough room
5512 6414 IF (.CHAR_STRING[SIZE] + (2 * K_PAD_LEN))
5513 6415 LEQ
5514 6416 (.LOC_FRAME_WIDTH * .LOC_FRAME_LENGTH)
5515 6417 THEN
5516 6418 BEGIN
5517 6419 ! Set the size to correct value
5518 6420 !
5519 6421 IF .CHAR_STRING[SIZE] LEQ (K_MAX_BUFFER_SIZE-2*(K_PAD_LEN)) THEN
5520 6422 STR_DESC[SIZE] = .CHAR_STRING[SIZE] + (2 * (K_PAD_LEN))
5521 6423 ELSE
5522 6424 STR_DESC[SIZE] = K_MAX_BUFFER_SIZE;
5523 6425 !
5524 6426 ! Pad the string using the local buffer
5525 6427 CH$COPY (K_PAD_LEN, PAD_CHAR, .CHAR_STRING[SIZE], .CHAR_STRING[ADDR],
5526 6428 K_PAD_LEN, PAD_CHAR, .PAD_CHAR, .STR_DESC[SIZE],
5527 6429 .STR_DESC[ADDR]);
5528 6430 END
5529 6431 ELSE ! copy into local buffer
5530 6432 BEGIN
5531 6433 CH$MOVE(.CHAR_STRING[SIZE], .CHAR_STRING[ADDR], .STR_DESC[ADDR]);
5532 6434 STR_DESC[SIZE] = .CHAR_STRING[SIZE];
5533 6435 END;
5534 6436 !
5535 6437 ! Calc offset to pointer using new padded length
5536 6438 IF (.LOC_FRAME_WIDTH * .LOC_FRAME_LENGTH) GTR (.STR_DESC[SIZE])
5537 6439 THEN
5538 6440 PTR_OFFSET = ((.LOC_FRAME_WIDTH * .LOC_FRAME_LENGTH)
```

```
5539 6441 2          - (.STR_DESC[SIZE]))/2
5540 6442 2 ELSE
5541 6443 2     PTR_OFFSET = 0;
5542 6444 2
5543 6445 2     ! Check for negative offset
5544 6446 2     IF .PTR_OFFSET LSS 0
5545 6447 2     THEN
5546 6448 2         PTR_OFFSET = 0;
5547 6449 2
5548 6450 2     ! Set pointer to buffer
5549 6451 2     STR_PTR = .PTR + .PTR_OFFSET;
5550 6452 2     STR_SIZE = .STR_DESC[SIZE];
5551 6453 2
5552 6454 2     DECRU L FROM .LOC_FRAME_LENGTH TO 1 DO
5553 6455 2         BEGIN
5554 6456 2
5555 6457 2         LOCAL NUM_CHARS; ! number of chars to move to the frame
5556 6458 2
5557 6459 2         IF .PTR_OFFSET GEQ .LOC_FRAME_WIDTH THEN
5558 6460 2             BEGIN
5559 6461 2                 PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH]; ! go to next row of frame
5560 6462 2                 PTR_OFFSET = .PTR_OFFSET - .LOC_FRAME_WIDTH; ! adjust offset to column
5561 6463 2             END
5562 6464 2         ELSE BEGIN
5563 6465 2
5564 6466 2             IF .STR_SIZE GEQ (.LOC_FRAME_WIDTH - .PTR_OFFSET) THEN
5565 6467 2                 NUM_CHARS = .LOC_FRAME_WIDTH - .PTR_OFFSET
5566 6468 2             ELSE ! check for overflow of frame width
5567 6469 2                 NUM_CHARS = .STR_SIZE; ! - insert which ever is less
5568 6470 2
5569 6471 2             CHSMOVE(.NUM_CHARS, .STR_DESC[ADDR], .STR_PTR);
5570 6472 2             STR_PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH]; ! Address calc. is base on
5571 6473 2             PTR = .STR_PTR; ! Frame ptr[0,0] & Form Width
5572 6474 2
5573 6475 2             STR_DESC[ADDR] = .STR_DESC[ADDR] + .NUM_CHARS;
5574 6476 2
5575 6477 2             IF .LOC_FRAME_WIDTH GTRU .STR_SIZE ! Already inserted it all
5576 6478 2             THEN
5577 6479 2                 EXITLOOP;
5578 6480 2
5579 6481 2             STR_SIZE = .STR_SIZE - .NUM_CHARS; ! Decrease string size
5580 6482 2
5581 6483 2             ! reset the offset to start column one
5582 6484 2             PTR_OFFSET = 0;
5583 6485 2
5584 6486 2         END;
5585 6487 2     END;
5586 6488 1 END;
```

```
OFFC 0000 CENTER_FRAME:
SE      FDE8  CE 9E 0002  .WORD  Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11  : 6360
52      14  AC DO 0007  MOVAB -536(SP), SP                      :
                               MOVL FRAME_LENGTH, R2                : 6383
```

				03	15	0000B	BLEQ	1\$		
				10	AC	D5 0000D	TSTL	FRAME_WIDTH		
					01	14 00010	BGTR	2\$		
						04 00012	RET			
		50		04	AC	D0 00013	MOVL	SCB, R0		6386
		51		01F8	C0	D0 00017	MOVL	504(R0), R1		
		51			52	D1 0001C	CMPL	R2, R1		
					06	14 0001F	BGTR	3\$		
	0200	C0		10	AC	D1 00021	CMPL	FRAME_WIDTH, 512(R0)		6387
					01	15 00027	BLEQ	4\$		
						04 00029	RET			
		5A		08	BC	D0 0002A	MOVL	@CHAR_STRING, R10		6392
					01	12 0002E	BNEQ	5\$		
						04 00030	RET			
		56			52	D0 00031	MOVL	R2, LOC_FRAME_LENGTH		6396
		51			56	D1 00034	CMPL	LOC_FRAME_LENGTH, R1		6397
					03	15 00037	BLEQ	6\$		
		56			51	D0 00039	MOVL	R1, LOC_FRAME_LENGTH		6399
		57		10	AC	D0 0003C	MOVL	FRAME_WIDTH, LOC_FRAME_WIDTH		6401
	04	AE		0200	C0	9E 00040	MOVAB	512(R0), 4(SP)		6402
	04	BE			57	D1 00046	CMPL	LOC_FRAME_WIDTH, @4(SP)		
					04	15 0004A	BLEQ	7\$		
		57		04	BE	D0 0004C	MOVL	@4(SP), LOC_FRAME_WIDTH		6404
	10	AE		0200	8F	3C 00050	MOVZWL	#512, STR_DESC		6408
	14	AE		18	AE	9E 00056	MOVAB	BUFFER, STR_DESC+4		6409
	0C	AE		2020	8F	3C 0005B	MOVZWL	#8224, PAD_CHAR		6410
	08	AE		0C	AC	D0 00061	MOVL	FRAME_PTR, PTR		6411
		58		08	AC	D0 00066	MOVL	CHAR_STRING, R8		6427
		50		04	AA	9E 0006A	MOVAB	4(R10), R0		6414
6E		57			56	C5 0006E	MULL3	LOC_FRAME_LENGTH, LOC_FRAME_WIDTH, (SP)		6416
		6E			50	D1 00072	CMPL	R0, (SP)		
					47	14 00075	BGTR	10\$		
	000001FC	8F			5A	D1 00077	CMPL	R10, #508		6421
					06	14 0007E	BGTR	8\$		
	10	AE			50	D0 00080	MOVL	R0, STR_DESC		6422
					06	11 00084	BRB	9\$		
	10	AE		0200	8F	3C 00086	MOVZWL	#512, STR_DESC		6424
		58		10	AE	D0 0008C	MOVL	STR_DESC, R11		6428
		59		14	AE	D0 00090	MOVL	STR_DESC+4, R9		6429
5B	0C	AE	0C	AE	02	2C 00094	MOVCS	#2, PAD_CHAR, PAD_CHAR, R11, (R9)		
					69	0009B				
					2A	18 0009C	BGEQ	11\$		
		59			02	C0 0009E	ADDL2	#2, R9		
		58			02	C2 000A1	SUBL2	#2, R11		
5B	0C	AE	04	B8	5A	2C 000A4	MOVCS	R10, @4(R8), PAD_CHAR, R11, (R9)		
					69	000AB				
					1A	18 000AC	BGEQ	11\$		
		59			5A	C0 000AE	ADDL2	R10, R9		
		58			5A	C2 000B1	SUBL2	R10, R11		
5B	0C	AE	0C	AE	02	2C 000B4	MOVCS	#2, PAD_CHAR, PAD_CHAR, R11, (R9)		
					69	000BB				
					0A	11 000BC	BRB	11\$		6414
	14	BE	04	B8	5A	28 000BE	MOVCS	R10, @4(R8), @STR_DESC+4		6433
			10	AE	5A	D0 000C4	MOVL	R10, STR_DESC		6434
			10	AE	6E	D1 000C8	CMPL	(SP), STR_DESC		6438
					0B	15 000CC	BLEQ	12\$		
	50		6E	10	AE	C3 000CE	SUBL3	STR_DESC, (SP), R0		6441

SEPARATE
V04-001

Print Symbiont -- separation routines
CENTER_FRAME - Insert String Information into t

H 11
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 185
(46)

5A	50	02	C7	000D3	DIVL3	#2, RO, PTR_OFFSET	6440
		02	11	000D7	BRB	13\$	6443
		5A	D4	000D9	CLRL	PTR_OFFSET	6446
		02	18	000DB	BGEQ	14\$	6448
		5A	D4	000DD	CLRL	PTR_OFFSET	6451
5B	5A	08	AE	C1	ADDL3	PTR, PTR_OFFSET, STR_PTR	6452
	59	10	AE	D0	MOVL	STR_DESC, STR_SIZE	6454
			3F	11	BRB	20\$	6459
	57		5A	D1	CMPL	PTR_OFFSET, LOC_FRAME_WIDTH	6461
			0A	19	BLSS	16\$	6462
	08	AE	BE	C0	ADDL2	@4(SP), PTR	6459
		5A	57	C2	SUBL2	LOC_FRAME_WIDTH, PTR_OFFSET	6466
			2E	11	BRB	19\$	6467
50	57		5A	C3	SUBL3	PTR_OFFSET, LOC_FRAME_WIDTH, RO	6469
	50		59	D1	CMPL	STR_SIZE, RO	6471
			05	19	BLSS	17\$	6472
	58		50	D0	MOVL	RO, NUM_CHARS	6473
			03	11	BRB	18\$	6475
	58		59	D0	MOVL	STR_SIZE, NUM_CHARS	6477
6B	14	BE	58	28	MOVCL3	NUM_CHARS, @STR_DESC+4, (STR_PTR)	6481
5B	08	AE	BE	C1	ADDL3	@4(SP), PTR, STR_PTR	6484
	08	AE	5B	D0	MOVL	STR_PTR, PTR	6488
	14	AE	58	C0	ADDL2	NUM_CHARS, STR_DESC+4	
		59	57	D1	CMPL	LOC_FRAME_WIDTH, STR_SIZE	
			0B	1A	BGTRU	21\$	
		59	58	C2	SUBL2	NUM_CHARS, STR_SIZE	
			5A	D4	CLRL	PTR_OFFSET	
			56	D7	DECL	L	
			56	D5	TSTL	L	
			BD	12	BNEQ	15\$	
			04	0012D	RET		

; Routine Size: 302 bytes, Routine Base: CODE + 2EF7

```
5588 6489 1 %sbttl 'MERGE_FRAME - Merge Information in this Frame of the Page'
5589 6490 1 ++
5590 6491 1 Functional Description:
5591 6492 1 This procedure merges rows of non-blank strings into an array(frame).
5592 6493 1 Merging continues until either no more strings or no more frame. The
5593 6494 1 contents of the frame are merged to the base of the frame. Any row
5594 6495 1 with data present is considered impure and is merged.
5595 6496 1
5596 6497 1 Formal Parameters:
5597 6498 1 SCB - Address of the SCB
5598 6499 1 CHAR_STRING - Descriptor of String to Insert
5599 6500 1 FRAME_PTR - Address of first byte of Frame
5600 6501 1 FRAME_LENGTH - Length of Frame
5601 6502 1 FRAME_WIDTH - Width of Frame
5602 6503 1
5603 6504 1 Implicit Inputs:
5604 6505 1 none
5605 6506 1
5606 6507 1 Implicit Outputs:
5607 6508 1 none
5608 6509 1
5609 6510 1 Returned Value:
5610 6511 1 none
5611 6512 1
5612 6513 1 Side Effects:
5613 6514 1 Truncation is possible.
5614 6515 1
5615 6516 1 --
5616 6517 1 ROUTINE MERGE_FRAME (
5617 6518 1 SCB : REF $BLOCK,
5618 6519 1 CHAR_STRING : REF VECTOR[2],
5619 6520 1 FRAME_PTR : REF PAGE_ARRAY,
5620 6521 1 FRAME_WIDTH , Number of Columns
5621 6522 1 FRAME_LENGTH : Number of Rows
5622 6523 1 RET_LEN : REF VECTOR[word]
5623 6524 1 ) : NOVALUE =
5624 6525 2 BEGIN
5625 6526 2
5626 6527 2 LITERAL K_MAX_SIZE = 256;
5627 6528 2
5628 6529 2 LOCAL
5629 6530 2 CLR_STR : VECTOR[2],
5630 6531 2 SRCE_STR : VECTOR[2],
5631 6532 2 BUFFER : VECTOR[256,byte],
5632 6533 2 CURRENT_PTR : REF PAGE_ARRAY,
5633 6534 2 CURRENT_LEN ,
5634 6535 2 DEST_OFFSET ,
5635 6536 2 SOURCE_OFFSET ;
5636 6537 2
5637 6538 2 ! setup clr_str
5638 6539 2
5639 6540 2 CLR_STR[ADDR] = .CHAR_STRING[ADDR];
5640 6541 2 CLR_STR[SIZE] = .FRAME_WIDTH;
5641 6542 2 CH$FILL ( '%', .FRAME_WIDTH, .CLR_STR[ADDR]);
5642 6543 2 ! setup source_str
5643 6544 2
5644 6545 2 SRCE_STR[SIZE] = %ALLOCATION(BUFFER);
```

```
5645 6546 2 SRCE_STR[ADDR] = BUFFER;
5646 6547
5647 6548 ! setup current point and offsets into frame
5648 6549
5649 6550 CURRENT_PTR = FRAME_PTR[0,0, .SCB[PSMSL_PAGE_WIDTH]];
5650 6551
5651 6552 SOURCE_OFFSET = .FRAME_LENGTH-1; ! start at frame boundaries
5652 6553 DEST_OFFSET = .FRAME_LENGTH-1;
5653 6554
5654 6555 DECRU I FROM (.FRAME_LENGTH-1) TO 0
5655 6556 DO
5656 6557 BEGIN
5657 6558 FIND_DEST_PTR ( .SCB,
5658 6559 CLR_STR[0],
5659 6560 CURRENT_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
5660 6561 .FRAME_WIDTH,
5661 6562 .FRAME_LENGTH,
5662 6563 DEST_OFFSET);
5663 6564
5664 6565 FIND_SOURCE_PTR ( .SCB,
5665 6566 SRCE_STR[0],
5666 6567 CURRENT_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
5667 6568 .FRAME_WIDTH,
5668 6569 DEST_OFFSET,
5669 6570 SOURCE_OFFSET);
5670 6571
5671 6572 ! Exit loop when no source string
5672 6573
5673 6574 IF .SRCE_STR[SIZE] EQL 0
5674 6575 THEN
5675 6576 BEGIN
5676 6577 RET_LEN[0] = .FRAME_LENGTH - .1;
5677 6578 RETURN;
5678 6579 END;
5679 6580
5680 6581 ! Move the source to the destination
5681 6582
5682 6583 MOVE_FRAME (
5683 6584 .SCB,
5684 6585 SRCE_STR[0], ! string frame reference
5685 6586 CURRENT_PTR[0,.DEST_OFFSET,.SCB[PSMSL_PAGE_WIDTH]], ! ref to frame
5686 6587 .FRAME_WIDTH, ! cols to fill
5687 6588 i); ! rows to fill
5688 6589
5689 6590 ! Clear the source position
5690 6591
5691 6592 MOVE_FRAME (
5692 6593 .SCB,
5693 6594 CLR_STR[0], ! string frame reference
5694 6595 CURRENT_PTR[0,.SOURCE_OFFSET,.SCB[PSMSL_PAGE_WIDTH]], ! ref to frame
5695 6596 .FRAME_WIDTH, ! cols to fill
5696 6597 i); ! rows to fill
5697 6598
5698 6599
5699 6600 SRCE_STR[SIZE] = K_MAX_SIZE;
5700 6601
5701 6602
```

SEPARATE
V04-001

: 5702
: 5703

Print Symbiont -- separation routines
MERGE_FRAME - Merge Information in this Frame o

K 11

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 188
(47)

6603 2 END;
6604 1 END;

				007C 00000 MERGE_FRAME:						
		5E	FEE8	CE	9E	00002	WORD	Save R2,R3,R4,R5,R6	6517	
		50	08	AC	DO	00007	MOVAB	-280(SP), SP		
	FC	AD	04	AO	DO	0000B	MOVL	CHAR_STRING, R0	6540	
		56	10	AC	DO	00010	MOVL	4(R0), CLR_STR+4		
	F8	AD		56	DO	00014	MOVL	FRAME_WIDTH, R6	6541	
56	20	6E		00	2C	00018	MOVL	R6, CLR_STR		
			FC	BD		0001D	MOVCS	#0, (SPT), #32, R6, @CLR_STR+4	6542	
		F0	AD	0100	BF	3C	0001F	MOVZWL	#256, SRCE_STR	6545
		F4	AD	08	AE	9E	00025	MOVAB	BUFFER, SRCE_STR+4	6546
			53	OC	AC	DO	0002A	MOVL	FRAME_PTR, CURRENT_PTR	6550
	52	14	AC	01	C3	0002E	SUBL3	#1, FRAME_LENGTH, R2	6552	
		04	AE	52	DO	00033	MOVL	R2, SOURCE_OFFSET		
			6E	52	DO	00037	MOVL	R2, DEST_OFFSET	6553	
			54	04	AC	DO	0003A	MOVL	SCB, R4	6558
				5E	DD	0003E	1\$: PUSHL	SP	6560	
				14	AC	DD	00040	PUSHL	FRAME_LENGTH	6562
			0048	BF	BB	00043	PUSHR	#M<R3,R6>	6560	
			F8	AD	9F	00047	PUSHAB	CLR_STR	6559	
				54	DD	0004A	PUSHL	R4	6560	
	0000V	CF		06	FB	0004C	CALLS	#6, FIND_DEST_PTR		
			04	AE	9F	00051	PUSHAB	SOURCE_OFFSET	6567	
			04	AE	DD	00054	PUSHL	DEST_OFFSET	6569	
			0048	BF	BB	00057	PUSHR	#M<R3,R6>	6567	
			F0	AD	9F	0005B	PUSHAB	SRCE_STR	6566	
				54	DD	0005E	PUSHL	R4	6567	
	0000V	CF		06	FB	00060	CALLS	#6, FIND_SOURCE_PTR		
			F0	AD	D5	00065	TSTL	SRCE_STR	6574	
				07	12	00068	BNEQ	2\$		
18	BC	14	AC	52	A3	0006A	SUBW3	1, FRAME_LENGTH, @RET_LEN	6577	
				04		00070	RET		6576	
				01	DD	00071	2\$: PUSHL	#1	6586	
				56	DD	00073	PUSHL	R6	6588	
	50	08	AE	0200	C4	C5	00075	MULL3	512(R4), DEST_OFFSET, R0	6586
				6043	9F	0007C	PUSHAB	(R0)[CURRENT_PTR]		
			F0	AD	9F	0007F	PUSHAB	SRCE_STR	6585	
				54	DD	00082	PUSHL	R4	6586	
	FCFF	CF		05	FB	00084	CALLS	#5, MOVE_FRAME		
				01	DD	00089	PUSHL	#1	6596	
				56	DD	0008B	PUSHL	R6	6598	
	50	0C	AE	0200	C4	C5	0008D	MULL3	512(R4), SOURCE_OFFSET, R0	6596
				6043	9F	00094	PUSHAB	(R0)[CURRENT_PTR]		
			F8	AD	9F	00097	PUSHAB	CLR_STR	6595	
				54	DD	0009A	PUSHL	R4	6596	
	FCE7	CF		05	FB	0009C	CALLS	#5, MOVE_FRAME		
	F0	AD	0100	BF	3C	000A1	MOVZWL	#256, SRCE_STR	6601	
				52	D7	000A7	DECL	1	6555	
				93	11	000A9	BRB	1\$		

SEPARATE
V04-001

Print Symbiont -- separation routines
MERGE_FRAME - Merge Information in this frame o

L 11
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 189
(47)

; Routine Size: 171 bytes, Routine Base: CODE + 3025

```
5705 6605 1 %sbttl 'INSERT_NAME_BANNER - Insert User Name as Banner into this Frame'
5706 6606 1 ++
5707 6607 1 Functional Description:
5708 6608 1 This procedure inserts a string into the center of an array(frame).
5709 6609 1 The workhorse of this routine is the BANNER routine which creates the
5710 6610 1 large letters. Insertion is attempted in the center of the frame.
5711 6611 1 Otherwise, insertion starts at the left margin until no more characters
5712 6612 1 will fit.
5713 6613 1 NOTE: Returns the amount of space used in the frame
5714 6614 1
5715 6615 1 Formal Parameters:
5716 6616 1 SCB - Address of the SCB
5717 6617 1 CHAR_DESC - Desc String to Insert
5718 6618 1 FRAME_PTR - Address of first byte of Frame
5719 6619 1 FRAME_LENGTH - Length of Frame and Largest Banner
5720 6620 1 FRAME_WIDTH - Width of Frame and Height of Characters
5721 6621 1 DESIRED_BAN_TYPE - Requested banner type
5722 6622 1
5723 6623 1 Implicit Inputs:
5724 6624 1 none
5725 6625 1
5726 6626 1 Implicit Outputs:
5727 6627 1 none
5728 6628 1
5729 6629 1 Returned Value:
5730 6630 1 none
5731 6631 1
5732 6632 1 Side Effects:
5733 6633 1 Truncation is possible.
5734 6634 1
5735 6635 1 --
5736 6636 1 ROUTINE INSERT_NAME_BANNER (
5737 6637 1 SCB : REF $BBLOCK,
5738 6638 1 CHAR_DESC : REF VECTOR[2],
5739 6639 1 FRAME_PTR : REF PAGE_ARRAY,
5740 6640 1 FRAME_WIDTH : , ! Number of Columns
5741 6641 1 FRAME_LENGTH : , ! Number of Rows
5742 6642 1 DESIRED_BAN_TYPE : , ! Banner size desired
5743 6643 1 ) =
5744 6644 2 BEGIN
5745 6645 2 Define literals to use in 'Banner' call
5746 6646 2 (incl... char_repeat, line_repeat, spacing)
5747 6647 2
5748 6648 2 LITERAL K_LARGE_LETTERS = 14, ! Double size chars
5749 6649 2 K_MAX_STRING_SIZE = 42, ! max expanded chars(512 buffer)
5750 6650 2 K_ALT_CHAR = 0, ! alternate construction char
5751 6651 2 K_SPACING = 2, ! between character spacing
5752 6652 2 K_LEAD_SPACES = 0, ! number of leading spaces
5753 6653 2 K_MAX_BUF = 512, ! max for this frame buffer
5754 6654 2 LEAD_MASK = %B'00100000', ! convert lower to upper case
5755 6655 2 TRAILING = 1, ! flag for discard
5756 6656 2 ! (anything but 0 is trailing)
5757 6657 2
5758 6658 2 By defining local buffer and descriptor. I can call the banner routine
5759 6659 2 and get the length of the string ... then use an algorithm to center the
5760 6660 2 string into the frame.
5761 6661 2
```

```
5762 6662 2 LOCAL
5763 6663 NUM_LINES
5764 6664 CHAR_REPEAT ; character repeat
5765 6665 LINE_REPEAT ; line repeat
5766 6666 BUFFER : VECTOR [512,byte], assume max size 512 bytes
5767 6667 STRING_DESC : VECTOR [2], descriptor to current string
5768 6668 STR_PTR ; temp addr of input string
5769 6669 STR_LEN ; temp length of input string
5770 6670 RET_LEN : VECTOR[1]; Return Length Used
5771 6671
5772 6672 ! Dont even try if no frame
5773 6673
5774 6674 IF (.FRAME_LENGTH LSS 7) ! won't ever fit !!
5775 6675 OR
5776 6676 (.FRAME_WIDTH LEQ 0) ! nadda...
5777 6677 THEN
5778 6678 RETURN 0; ! dont even try... no room...
5779 6679
5780 6680 STR_LEN = .CHAR_DESC[SIZE]; ! move them into registers
5781 6681 STR_PTR = .CHAR_DESC[ADDR];
5782 6682
5783 6683 BASEEDIT (CHAR_DESC[0], CHAR_DESC[0], UPCASE_MASK);
5784 6684 ! lower to upper case character
5785 6685 ! Insert only the string ... No trailing blanks
5786 6686
5787 6687 DISCARD (TRAILING, XC' ', .STR_PTR, .STR_LEN,
5788 6688 STR_LEN, STR_PTR); ! Return length and pointer
5789 6689
5790 6690
5791 6691 ! init the character spacing ...depends on frame_length passed in !
5792 6692 assume small chars.
5793 6693 CHAR_REPEAT = 1; ! times to repeat a char
5794 6694 LINE_REPEAT = 1; ! times to repeat a line
5795 6695 NUM_LINES = 7; ! lines equal height of banner
5796 6696
5797 6697 ! Attempt to give the caller what he wants
5798 6698 Only use large banners if they fit in the frame
5799 6699
5800 6700 IF (
5801 6701 (.DESIRED_BAN_TYPE EQL K_LARGE_LETTERS) ! if he wants it
5802 6702 AND
5803 6703 ((.FRAME_LENGTH GEQ K_LARGE_LETTERS) ! and...
5804 6704 AND ! if there is room !!!
5805 6705 ((.FRAME_WIDTH/12) GEQ .STR_LEN))
5806 6706
5807 6707 THEN
5808 6708 BEGIN
5809 6709 CHAR_REPEAT = 2; ! times to repeat a char
5810 6710 LINE_REPEAT = 2; ! times to repeat a line
5811 6711 NUM_LINES = 14; ! height of banner
5812 6712 END;
5813 6713
5814 6714 ! truncate long character names to fit in buffer
5815 6715
5816 6716 IF .STR_LEN GTR K_MAX_STRING_SIZE
5817 6717 THEN
5818 6718 STR_LEN = K_MAX_STRING_SIZE; ! maximum 42 chars in
```

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_NAME_BANNER - Insert User Name as Banner

B 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 192
(48)

```
.. 5819 6719 2 ! buffer of 512
5820 6720 2 ! get the buffer
5821 6721 2
5822 6722 2 STRING_DESC[SIZE] = %ALLOCATION(BUFFER);
5823 6723 2 STRING_DESC[ADDR] = BUFFER;
5824 6724 2
5825 6725 2 INCR LINE_NO FROM 0 TO (.NUM_LINES - 1)
5826 6726 2 DO
5827 6727 2 BEGIN
5828 6728 2 PSM$BANNER (.SCB,
5829 6729 2 .STR_LEN,
5830 6730 2 .STR_PTR,
5831 6731 2 K_LEAD_SPACES,
5832 6732 2 .CHAR_REPEAT,
5833 6733 2 .LINE_REPEAT,
5834 6734 2 K_SPACING,
5835 6735 2 K_ALT_CHAR,
5836 6736 2 .STRING_DESC[ADDR],
5837 6737 2 .LINE_NO,
5838 6738 2 STRING_DESC[SIZE]);
5839 6739 2
5840 6740 2 CENTER_FRAME ( .SCB,
5841 6741 2 STRING_DESC[0],
5842 6742 2 FRAME_PTR[0, .LINE_NO, .SCB[PSM$L_PAGE_WIDTH]],
5843 6743 2 .FRAME_WIDTH, 1);
5844 6744 2 ! re-init
5845 6745 2 STRING_DESC[SIZE] = K_MAX_BUF; ! reset buffer size
5846 6746 2 END;
5847 6747 2
5848 6748 2 RETURN .NUM_LINES
5849 6749 1 END;
```

```
007C 00000 INSERT_NAME BANNER:
          5E FDF0 CE 9E 00002 .WORD Save R2,R3,R4,R5,R6
          07 14 AC D1 00007 MOVAB -528(SP), SP
          05 19 0000B CMPL FRAME_LENGTH, #7
          10 AC D5 0000D BLSS 1$
          03 14 00010 TSTL FRAME_WIDTH
          00BD 31 00012 1$: BGTR 2$
          08 AC D0 00015 2$: BRW 7$
          04 50 08 60 D0 00019 MOVL CHAR_DESC, R0
          AE 04 A0 D0 0001D MOVL (R0), STR_LEN
          6E 20 DD 00021 MOVL 4(R0), STR_PTR
          50 DD 00023 PUSHL #32
          50 DD 00025 PUSHL R0
          03 FB 00027 PUSHL R0
          00000000G 00 03 FE 0002E CALLS #3, BASSEDT
          08 AE 9F 00030 PUSHL SP
          0C AE DD 00033 PUSHAB STR_LEN
          0C AE DD 00036 PUSHL STR_LEN
          20 DD 00039 PUSHL STR_PTR
          01 DD 0003B PUSHL #32
          PUSHL #1
```

6636
6674
6676
6680
6681
6683
6687

0000V	CF	06	FB	0003D	CALLS	#6, DISCARD	...	
	56	01	D0	00042	MOVL	#1, CHAR_REPEAT	...	6693
	55	01	D0	00045	MOVL	#1, LINE_REPEAT	...	6694
	54	07	D0	00048	MOVL	#7, NUM_LINES	...	6695
	0E	18	AC	D1 0004B	CMPL	DESIRED_BAN_TYPE, #14	...	6701
			1A	12 0004F	BNEQ	3\$...	
	0E	14	AC	D1 00051	CMPL	FRAME_LENGTH, #14	...	6703
			14	19 00055	BLSS	3\$...	
50	10	AC	0C	C7 00057	DIVL3	#12, FRAME_WIDTH, R0	...	6705
	04	AE	50	D1 0005C	CMPL	R0, STR_LEN	...	
			09	19 00060	BLSS	3\$...	
	56		02	D0 00062	MOVL	#2, CHAR_REPEAT	...	6709
	55		02	D0 00065	MOVL	#2, LINE_REPEAT	...	6710
	54		0E	D0 00068	MOVL	#14, NUM_LINES	...	6711
	2A	04	AE	D1 0006B	CMPL	STR_LEN, #42	...	6716
			04	15 0006F	BLEQ	4\$...	
	04	AE	2A	D0 00071	MOVL	#42, STR_LEN	...	6718
	08	AE	8F	3C C0075	MOVZWL	#512, STRING_DESC	...	6722
	0C	AE	10	9E 0007B	MOVAB	BUFFER, STRING_DESC+4	...	6723
			04	AC D0 00080	MOVL	SCB, R3	...	6742
	53		01	CE 00084	MNEGL	#1, LINE_NO	...	
	52		41	11 00087	BRB	6\$...	
		08	AE	9F 00089	PUSHAB	STRING_DESC	...	6738
			52	DD 0008C	PUSHL	LINE_NO	...	6737
		14	AE	DD 0008E	PUSHL	STRING_DESC+4	...	6736
	7E		02	7D 00091	MOVQ	#2, -(SP)	...	6728
			55	DD 00094	PUSHL	LINE_REPEAT	...	6733
			56	DD 00096	PUSHL	CHAR_REPEAT	...	6732
			7E	D4 00098	CLRL	-(SP)	...	6728
		20	AE	DD 0009A	PUSHL	STR_PTR	...	6730
		28	AE	DD 0009D	PUSHL	STR_LEN	...	6729
		04	AC	DD 000A0	PUSHL	SCB	...	6728
00000000G	00		0B	FB 000A3	CALLS	#11, PSMSBANNER	...	
			01	DD 000AA	PUSHL	#1	...	6742
		10	AC	DD 000AC	PUSHL	FRAME_WIDTH	...	6743
50	52	0200	C3	C5 000AF	MULL3	512(R3), LINE_NO, R0	...	6742
		0C	BC40	9F 000B5	PUSHAB	@FRAME_PTR[R0]	...	
		14	AE	9F 000B9	PUSHAB	STRING_DESC	...	6741
		04	AC	DD 000BC	PUSHL	SCB	...	6742
	FD63	CF	05	FB 000BF	CALLS	#5, CENTER_FRAME	...	
	08	AE	8F	3C 000C4	MOVZWL	#512, STRING_DESC	...	6745
BB			54	F2 000CA	AOBLSS	NUM_LINES, LINE_NO, 5\$...	6725
			54	D0 000CE	MOVL	NUM_LINES, R0	...	6748
			J4	000D1	RET		...	
			50	D4 000D2	CLRL	R0	...	6749
			04	000D4	RET		...	

; Routine Size: 213 bytes, Routine Base: CODE + 30D0

```
5851 6750 1 %sbttl 'FIND_DEST_PTR - Finds an empty Position in the Frame'
5852 6751 1 ++
5853 6752 1 Functional Description:
5854 6753 1 This routine finds the first empty frame position from the bottom
5855 6754 1 of the frame. The returned parameters include the dest_ptr (position
5856 6755 1 found) and the length left in the frame. If unable to find an empty
5857 6756 1 position then RET_OFFSET = .FRAME_LENGTH
5858 6757 1
5859 6758 1 Formal Parameters:
5860 6759 1 SCB - Address of the SCB
5861 6760 1 CLR_STR - Descriptor of blank string
5862 6761 1 FRAME_PTR - Address of first byte of frame
5863 6762 1 FRAME_LENGTH - Length of frame and Largest Banner
5864 6763 1 FRAME_WIDTH - Width of frame and Height of Characters
5865 6764 1 RETURN_PTR - Pointer to position in frame
5866 6765 1 RET_LEN - Resultant length of frame
5867 6766 1
5868 6767 1 Implicit Inputs:
5869 6768 1 none
5870 6769 1
5871 6770 1 Implicit Outputs:
5872 6771 1 none
5873 6772 1
5874 6773 1 Returned Value:
5875 6774 1 none
5876 6775 1
5877 6776 1 Side Effects:
5878 6777 1 Truncation is possible.
5879 6778 1
5880 6779 1 --
5881 6780 1 ROUTINE FIND_DEST_PTR (
5882 6781 1 SCB : REF $BLOCK,
5883 6782 1 CLR_STR : REF VECTOR[2],
5884 6783 1 FRAME_PTR : REF PAGE_ARRAY,
5885 6784 1 FRAME_WIDTH : , Number of Columns
5886 6785 1 FRAME_LENGTH : , Number of Rows
5887 6786 1 RET_OFFSET : REF VECTOR
5888 6787 1 ) : NOVALUE =
5889 6788 2 BEGIN
5890 6789 2 LOCAL
5891 6790 2 CURR_PTR :
5892 6791 2
5893 6792 2 ! exit if frame length is zero
5894 6793 2 !
5895 6794 2 IF .FRAME_LENGTH EQL 0
5896 6795 2 THEN
5897 6796 2 RETURN;
5898 6797 2
5899 6798 2 DECR I FROM (.FRAME_LENGTH-1) TO 0
5900 6799 2 DO
5901 6800 2 BEGIN
5902 6801 2 CURR_PTR = FRAME_PTR[0, .I, .SCB[PSM$PAGE_WIDTH]];
5903 6802 2
5904 6803 2 IF CH$EQL( .FRAME_WIDTH, .CURR_PTR,
5905 6804 2 1, CH$PTR( UPLIT ('-')), %C' ' )
5906 6805 2 THEN
5907 6806 2 BEGIN
```

SEPARATE
V04-001

Print Symbiont -- separation routines
FIND_DEST_PTR - Finds an empty Position in the

E 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 195
(49)

```

: 5908      6807 4      RET OFFSET[0] = .I;
: 5909      6808 4      RETURN;
: 5910      6809 3      END;
: 5911      6810 2      END;
: 5912      6811 2
: 5913      6812 2      RET_OFFSET[0] = 0;
: 5914      6813 2
: 5915      6814 1      END;

```

! exit this routine

! exit with 0 if destination not found

00 00 00 20 031A5 .BLKB 3
031A8 P.AGN: .ASCII \ \<0><0><0>

		007C 00000 FIND_DEST_PTR:				
		14	AC	D5	00002	WORD Save R2,R3,R4,R5,R6
			2A	13	00005	TSTL FRAME_LENGTH
	55	04	AC	D0	00007	BEQL 3\$
	54	14	AC	D0	0000B	MOVL SCB, R5
			1A	11	0000F	MOVL FRAME_LENGTH, 1
						BRB 2\$
	50	54	0200	C5	C5 00011	MULL3 512(R5), 1, R0
	56	50	0C	AC	C1 00017	ADDL3 FRAME_PTR, R0, CURR_PTR
01	20	66	10	AC	2D 0001C	CMPC5 FRAME_WIDTH, (CURR_PTR), #32, #1, P.AGN
			D8	AF		
			05	12	00024	BNEQ 2\$
	18	BC		54	D0 00026	MOVL 1, @RET_OFFSET
				04	0002A	RET
	E3			54	F4 0002B	SOBGEQ 1, 1\$
		18	BC	D4	0002E	CLRL @RET_OFFSET
				04	00031	RET

6780
6794
6801
6803
6801
6803
6807
6806
6798
6812
6814

; Routine Size: 50 bytes, Routine Base: CODE + 31AC

```
5917 6815 1 %sbttl 'FIND_SOURCE_PTR - Finds an empty Position in the frame'
5918 6816 1 ++
5919 6817 1 Functional Description:
5920 6818 1 This routine finds the first nonempty frame position from the bottom
5921 6819 1 of the frame. The returned parameters include the ret_str descriptor
5922 6820 1 return_ptr (position found) and the length left in the frame. If
5923 6821 1 unable to find a string position then RET_STR[SIZE]=0, RET_LEN = 0 and
5924 6822 1 RETURN_PTR = FRAME_PTR.
5925 6823 1
5926 6824 1 Formal Parameters:
5927 6825 1 SCB - Address of the SCB
5928 6826 1 RET_STR - Descriptor of buffer for return string
5929 6827 1 FRAME_PTR - Address of first byte of frame
5930 6828 1 FRAME_LENGTH - Length of frame and Largest Banner
5931 6829 1 FRAME_WIDTH - Width of frame and Height of Characters
5932 6830 1 RET_OFFSET - Pointer to position in frame
5933 6831 1
5934 6832 1 Implicit Inputs:
5935 6833 1 none
5936 6834 1
5937 6835 1 Implicit Outputs:
5938 6836 1 none
5939 6837 1
5940 6838 1 Returned Value:
5941 6839 1 none
5942 6840 1
5943 6841 1 Side Effects:
5944 6842 1 Truncation is possible.
5945 6843 1
5946 6844 1 --
5947 6845 1 ROUTINE FIND_SOURCE_PTR (
5948 6846 1 SCB : REF $BLOCK,
5949 6847 1 RET_STR : REF VECTOR[2],
5950 6848 1 FRAME_PTR : REF PAGE_ARRAY,
5951 6849 1 FRAME_WIDTH : , ! Number of Columns
5952 6850 1 FRAME_LENGTH : , ! Number of Rows
5953 6851 1 RET_OFFSET : REF VECTOR
5954 6852 1 ) : NOVALUE =
5955 6853 2 BEGIN
5956 6854 2
5957 6855 2 LOCAL
5958 6856 2 CURR_PTR;
5959 6857 2
5960 6858 2
5961 6859 2 ! exit if frame length is zero
5962 6860 2
5963 6861 2 IF .FRAME_LENGTH EQL 0
5964 6862 2 THEN
5965 6863 2 BEGIN
5966 6864 2 RET_STR[SIZE] = 0;
5967 6865 2 RETURN;
5968 6866 2 END;
5969 6867 2
5970 6868 2 DECR I FROM (.FRAME_LENGTH-1) TO 0
5971 6869 2 DO
5972 6870 2 BEGIN
5973 6871 2 CURR_PTR = FRAME_PTR[0..I..SCB[PSM$L_PAGE_WIDTH]];
```


SEPARATE
V04-001

Print Symbiont -- separation routines
FIND_SOURCE_PTR - Finds an empty Position in th

G 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 197
(50)

```

5974 6872 3
5975 6873 3 IF CH$NEQ( .FRAME_WIDTH, .CURR_PTR,
5976 6874 3 1, CH$PTR( UPLIT ('-'), %C' ' )
5977 6875 3 THEN
5978 6876 3 BEGIN
5979 6877 3 CH$MOVE( .FRAME_WIDTH, .CURR_PTR,
5980 6878 3 .RET_STR[ADDR]);
5981 6879 3 RET_STR[SIZE] = .FRAME_WIDTH;
5982 6880 3 RET_OFFSET[0] = .1;
5983 6881 3 RETURN; ! exit this routine
5984 6882 3 END;
5985 6883 3 END;
5986 6884 3
5987 6885 2 RET_STR[SIZE] = 0;
5988 6886 2 RET_OFFSET = 0;
5989 6887 2
5990 6888 1 END;
```

00 00 00 20 031DE .BLKB 2
031E0 P.AGO: .ASCII \ \<0><0><0>

				03FC 00000 FIND_SOURCE_PTR:		
				14 AC D5 00002	WORD Save R2,R3,R4,R5,R6,R7,R8,R9	6845
				04 12 00005	TSTL FRAME_LENGTH	6861
				08 BC D4 00007	BNEQ 1\$	
				04 0000A	CLRL @RET_STR	6864
				58 04 AC D0 0000B 1\$:	RET	6863
				56 14 AC D0 0000F	MOVL SCB, R8	6871
				28 11 00013	MOVL FRAME_LENGTH, I	6873
				56 0200 C8 C5 00015 2\$:	BRB 3\$	
	50			59 0C AC C1 0001B	MULL3 512(R8), I, R0	6871
01	20			69 10 AC 2D 00020	ADDL3 FRAME_PTR, R0, CURR_PTR	
				D4 AF 00026	CMPC5 FRAME_WIDTH, (CURR_PTR), #32, #1, P.AGO	6873
				13 13 00028		
				57 08 AC D0 0002A	BEQL 3\$	
	04	B7		69 10 AC 28 0002E	MOVL RET_STR, R7	6878
				67 10 AC D0 00034	MOVC3 FRAME_WIDTH, (CURR_PTR), @4(R7)	
				18 BC 56 D0 00038	MOVL FRAME_WIDTH, (R7)	6879
				04 0003C	MOVL I, @RET_OFFSET	6880
				D5 56 F4 0003D 3\$:	RET	6876
				08 BC D4 00040	SOBGEQ I, 2\$	6868
				18 AC D4 00043	CLRL @RET_STR	6885
				04 00046	CLRL RET_OFFSET	6886
					RET	6888

; Routine Size: 71 bytes, Routine Base: CODE + 31E4

```
5992 6889 1 %sbttl 'DELIMIT_STRING - Return the last position of this delimiter'
5993 6890 1 ++
5994 6891 1 Functional Description:
5995 6892 1 This procedure returns the position of the delimiter nearest the
5996 6893 1 string_end
5997 6894 1
5998 6895 1 Formal Parameters:
5999 6896 1 STR_PTR - Pointer of String to delimit
6000 6897 1 CHAR - Character delimiter
6001 6898 1 STR_END - End position of string
6002 6899 1
6003 6900 1 Implicit Inputs:
6004 6901 1 none
6005 6902 1
6006 6903 1 Implicit Outputs:
6007 6904 1 none
6008 6905 1
6009 6906 1 Returned Value:
6010 6907 1 none
6011 6908 1
6012 6909 1 Side Effects:
6013 6910 1 none
6014 6911 1 --
6015 6912 1 ROUTINE DELIMIT_STRING (
6016 6913 1 STR_PTR ,
6017 6914 1 CHAR ,
6018 6915 1 STR_END ) =
6019 6916 2 BEGIN
6020 6917 2 LOCAL
6021 6918 2 POS,
6022 6919 2 BASE,
6023 6920 2 TEMP_PTR,
6024 6921 2 CHAR_PTR;
6025 6922 2
6026 6923 2 TEMP_PTR = .STR_PTR;
6027 6924 2 BASE = .STR_PTR + .STR_END - 1;
6028 6925 2 CHAR_PTR = CH$PTR(CHAR);
6029 6926 2
6030 6927 2 DECR CURR_PTR FROM (.BASE) TO .STR_PTR DO
6031 6928 2 BEGIN
6032 6929 2 TEMP_PTR = CH$PTR(.CURR_PTR);
6033 6930 2
6034 6931 2 POS = CH$EQL(1, .TEMP_PTR, 1, .CHAR_PTR);
6035 6932 2
6036 6933 2 IF (.POS EQL 1) AND
6037 6934 2 (.CURR_PTR EQL .STR_PTR)
6038 6935 2 THEN
6039 6936 2 RETURN .STR_END; ! ...return the original length
6040 6937 2
6041 6938 2 IF (.POS EQL 1) AND
6042 6939 2 (.CURR_PTR GTR .STR_PTR)
6043 6940 2 THEN ! char in string
6044 6941 2 RETURN (.CURR_PTR - .STR_PTR + 1); ! ...return position plus one
6045 6942 2
6046 6943 2 IF (.POS EQL 0) AND (.CURR_PTR EQL .STR_PTR) THEN
6047 6944 2 RETURN .STR_END; ! handle extra decrem
6048 6945 2 END;
```

SEPARATE
V04-001

Print Symbiont -- separation routines
DELIMIT_STRING - Return the last position of th

1 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 199
(51)

```
: 6049      6946  2
: 6050      6947  2 RETURN .STR_END;      ! ...return the original length
: 6051      6948  2
: 6052      6949  1 END;
```

```
                                003C 00000 DELIMIT_STRING:
                                .WORD Save R2,R3,R4,R5
52      04 AC D0 00002          MOVL STR_PTR, R2      : 6912
54      52 D0 00006          MOVL R2, TEMP_PTR      : 6923
50      52 0C AC C1 00009      ADDL3 STR_END, R2, R0  : 6924
55      50 D7 0000E          DECL BASE
55      08 AC 9E 00010          MOVAB CHAR, CHAR_PTR  : 6925
54      3A 11 00014          BRB 6$                : 6931
54      50 D0 00016 1$:      MOVL CURR_PTR, TEMP_PTR  : 6929
65      51 D4 00019          CLRL R1                : 6931
65      64 91 0001B          CMPB (TEMP_PTR), (CHAR_PTR)
53      02 12 0001E          BNEQ 2$
53      51 D6 00020          INCL R1
53      51 D0 00022 2$:      MOVL R1, POS
01      51 D4 00025          CLRL R1                : 6933
53      53 D1 00027          CMPL POS, #1
52      07 12 0002A          BNEQ 3$
52      51 D6 0002C          INCL R1
52      50 D1 0002E          CMPL CURR_PTR, R2        : 6934
52      22 13 00031          BEQL 7$
52      0F 51 E9 00033 3$:   BLBC R1, 4$            : 6938
52      50 D1 00036          CMPL CURR_PTR, R2        : 6939
51      0A 15 00039          BLEQ 4$
51      50 52 C3 0003B          SUBL3 R2, CURR_PTR, R1 : 6941
51      51 D6 0003F          INCL R1
50      51 D0 00041          MOVL R1, R0
50      04 00044          RET
53      53 D5 00045 4$:     TSTL POS                : 6943
52      05 12 00047          BNEQ 5$
52      50 D1 00049          CMPL CURR_PTR, R2
52      07 13 0004C          BEQL 7$
52      50 D7 0004E 5$:     DECL CURR_PTR            : 6927
52      50 D1 00050 6$:     CMPL CURR_PTR, R2
50      C1 18 00053          BGEQ 1$
50      0C AC D0 00055 7$:   MOVL STR_END, R0
50      04 00059          RET                        : 6947
: 6949
```

; Routine Size: 90 bytes, Routine Base: CODE + 3228

```
6054 6950 1 %sbttl 'DELIMIT_STRING_NOT - Return the last position of not this delimiter'
6055 6951 1 ++
6056 6952 1 Functional Description:
6057 6953 1 This procedure returns the length of the string without the delimited
6058 6954 1 characters on the string end. Return the original length if
6059 6955 1 non_delimiters cannot be found.
6060 6956 1
6061 6957 1 Formal Parameters:
6062 6958 1 STR_PTR - Pointer of String to delimit
6063 6959 1 CHAR - Character delimiter
6064 6960 1 STR_END - End position of string
6065 6961 1
6066 6962 1 Implicit Inputs:
6067 6963 1 none
6068 6964 1
6069 6965 1 Implicit Outputs:
6070 6966 1 none
6071 6967 1
6072 6968 1 Returned Value:
6073 6969 1 none
6074 6970 1
6075 6971 1 Side Effects:
6076 6972 1 none
6077 6973 1 --
6078 6974 1 ROUTINE DELIMIT_STRING_NOT(
6079 6975 1 STR_PTR,
6080 6976 1 CHAR,
6081 6977 1 STR_END ) =
6082 6978 2 BEGIN
6083 6979 2
6084 6980 2
6085 6981 2 LOCAL
6086 6982 2 PTR : REF VECTOR[,byte];
6087 6983 2
6088 6984 2 IF .STR_END EQL 0
6089 6985 2 THEN
6090 6986 2 RETURN 0;
6091 6987 2
6092 6988 2 PTR = .STR_PTR + .STR_END - 1;
6093 6989 2
6094 6990 2 WHILE .PTR GTRU .STR_PTR
6095 6991 2 DO
6096 6992 2 IF .PTR[0] NEQU .CHAR
6097 6993 2 THEN
6098 6994 2 EXITLOOP
6099 6995 2 ELSE
6100 6996 2 PTR = .PTR - 1;
6101 6997 2
6102 6998 2 RETURN .PTR - .STR_PTR + 1;
6103 6999 2
6104 7000 1 END;
```

0000 00000 DELIMIT_STRING_NOT:

SEPARATE
V04-001

Print Symbiont -- separation routines
DELIMIT_STRING_NOT - Return the last position o

K 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 201
(52)

			0C	AC	D5	00002		.WORD	Save nothing	:	6974
				1D	13	00005		TSTL	STR_END	:	6984
	50	04	AC	0C	AC	C1	00007	BEQL	3\$:	
					50	D7	0000D	ADDL3	STR_END, STR_PTR, R0	:	6988
		04	AC		50	D1	0000F	DECL	PTR	:	
					08	1B	00013	CMPL	PTR, STR_PTR	:	6990
08	AC	60		08	00	ED	00015	BLEQU	2\$:	
					FO	13	0001B	CMPZV	#0, #8, (PTR), CHAR	:	6992
				50	04	AC	C2	0001D	BEQL	1\$	
					50	D6	00021	SUBL2	STR_PTR, R0	:	6998
						04	00023	INCL	R0	:	
					50	D4	00024	RET		:	7000
					04	00026		CLRL	R0	:	
								RET		:	

; Routine Size: 39 bytes, Routine Base: CODE + 3285

```

6106 7001 1 %sbttl 'DISCARD - Returns a Pointer to First Char NOT Discarded'
6107 7002 1 ++
6108 7003 1 Functional Description:
6109 7004 1 This routine discards the character in the string from the beginning
6110 7005 1 (LEADING) or end(TRAILING) of the string(STR_PTR) and returns a pointer
6111 7006 1 to the first position that is found not to contain the discard
6112 7007 1 character(CHAR). The boundaries of the string are the beginning pointer
6113 7008 1 and the string length(LEN).
6114 7009 1
6115 7010 1 Formal Parameters:
6116 7011 1 WHICH_WAY - Leading/Trailing
6117 7012 1 CHAR - Character to discard
6118 7013 1 STR_PTR - Pointer of String to delimit
6119 7014 1 LEN - Length of string
6120 7015 1 RET_PTR - Return pointer to first undiscarded position
6121 7016 1
6122 7017 1 Implicit Inputs:
6123 7018 1 none
6124 7019 1
6125 7020 1 Implicit Outputs:
6126 7021 1 none
6127 7022 1
6128 7023 1 Returned Value:
6129 7024 1 none
6130 7025 1
6131 7026 1 Side Effects:
6132 7027 1 none
6133 7028 1 --
6134 7029 1 ROUTINE DISCARD (
6135 7030 1 WHICH_WAY,
6136 7031 1 CHAR,
6137 7032 1 STR_PTR,
6138 7033 1 LEN : WORD, ! force word size
6139 7034 1 RET_LEN : REF VECTOR,
6140 7035 1 RET_PTR : REF VECTOR) : NOVALUE =
6141 7036 2 BEGIN
6142 7037 2
6143 7038 2 LITERAL
6144 7039 2 LEADING = 0;
6145 7040 2
6146 7041 2 LOCAL
6147 7042 2 DSTR_DESC : VECTOR[2];
6148 7043 2
6149 7044 2 ! a descriptor is needed for BAS$EDIT routine
6150 7045 2 DSTR_DESC[SIZE] = .LEN;
6151 7046 2 DSTR_DESC[ADDR] = .STR_PTR;
6152 7047 2
6153 7048 2 IF .WHICH_WAY EQL LEADING THEN
6154 7049 2 BEGIN
6155 7050 2 BAS$EDIT ( DSTR_DESC[0], DSTR_DESC[0], LEAD_MASK); ! trim leading blanks
6156 7051 2 RET_PTR[0] = .DSTR_DESC[ADDR];
6157 7052 2 RET_LEN[0] = .DSTR_DESC[SIZE];
6158 7053 2 END
6159 7054 2 ELSE ! trim trailing blanks
6160 7055 2 BEGIN
6161 7056 2 RET_LEN[0] = DELIMIT_STRING_NOT ( .STR_PTR, .CHAR, .LEN);
6162 7057 2 RET_PTR[0] = .STR_PTR;
```

SEPARATE
V04-001

Print Symbiont -- separation routines
DISCARD - Returns a Pointer to First Char NOT D

M 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 203
(53)

: 6163
: 6164

7058 2 END;
7059 1 END;

				0000	00000	DISCARD: .WORD	Save nothing	: 7029
	5E		04	C2	00002	SUBL2	#4, SP	
	7E		10	AC	3C	MOVZWL	LEN, DSTR_DESC	: 7045
04	AE		0C	AC	D0	MOVL	STR_PTR, DSTR_DESC+4	: 7046
			04	AC	D5	TSTL	WHICH_WAY	: 7048
				19	12	BNEQ	1\$	
				08	DD	PUSHL	#8	: 7050
			04	AE	9F	PUSHAB	DSTR_DESC	
			08	AE	9F	PUSHAB	DSTR_DESC	
00000000G	00			03	FB	CALLS	#3, BASSEDIT	
18	BC		04	AE	D0	MOVL	DSTR_DESC+4, @RET_PTR	: 7051
14	BC			6E	D0	MOVL	DSTR_DESC, @RET_LEN	: 7052
				04	0002B	RET		: 7048
	7E		10	AC	3C	MOVZWL	LEN, -(SP)	: 7056
			08	AC	DD	PUSHL	CHAR	
			0C	AC	DD	PUSHL	STR_PTR	
9F	AF			03	FB	CALLS	#3, DELIMIT_STRING_NOT	
14	BC			50	D0	MOVL	R0, @RET_LEN	
18	BC		0C	AC	D0	MOVL	STR_PTR, @RET_PTR	: 7057
				04	00043	RET		: 7059

; Routine Size: 68 bytes, Routine Base: CODE + 32AC

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_OPEN - Boolean Valued routine indicating f

N 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 204
(54)

```
: 6166 7060 1 %sbttl 'FILE_OPEN - Boolean Valued routine indicating file open status'
: 6167 7061 1 ++
: 6168 7062 1 Functional Description:
: 6169 7063 1 This routine interrogates the FAB and determines if the current file
: 6170 7064 1 is open and/or if information can be extracted from the file.
: 6171 7065 1 TRUE = 1, FALSE = 0;
: 6172 7066 1
: 6173 7067 1 Formal Parameters:
: 6174 7068 1 none
: 6175 7069 1
: 6176 7070 1 Implicit Inputs:
: 6177 7071 1 none
: 6178 7072 1
: 6179 7073 1 Implicit Outputs:
: 6180 7074 1 none
: 6181 7075 1
: 6182 7076 1 Returned Value:
: 6183 7077 1 none
: 6184 7078 1
: 6185 7079 1 Side Effects:
: 6186 7080 1 none
: 6187 7081 1 --
: 6188 7082 1 ROUTINE FILE_OPEN (
: 6189 7083 1 SCB = : REF $BBLOCK ! SCB
: 6190 7084 1 )
: 6191 7085 2 BEGIN
: 6192 7086 2
: 6193 7087 2 RETURN .SCB[PSM$V_FAB_VALID]
: 6194 7088 2
: 6195 7089 1 END;
```

0000 00000 FILE_OPEN:

50	10	A0	50	04	AC	D0	00002	.WORD	Save nothing	
			01		04	EF	00006	MOVL	SCB, R0	: 7082
						04	0000C	EXTZV	#4, #1, 16(R0), R0	: 7087
								RET		: 7089

; Routine Size: 13 bytes, Routine Base: CODE + 32F0

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_OPEN - Boolean Valued routine indicating f

B 13
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 205
(55)

: 6197 7090 1 END
: 6198 7091 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
DATA	4	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
CODE	13053	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	88	0	1000	00:01.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SEPARATE/OBJ=OBJ\$:SEPARATE MSRC\$:SEPARATE/UPDATE=(ENH\$:SEPARATE)

: Size: 10301 code + 2756 data bytes
: Run Time: 03:23.1
: Elapsed Time: 06:52.7
: Lines/CPU Min: 2094
: Lexemes/CPU-Min: 22941
: Memory Used: 682 pages
: Compilation Complete

0310

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0311

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY